OZAmiga

Magazine

REAL 3D
Version 2

The *OZAmiga* is produced entirely on an A2000, using Pagestream 2.2.

Striving to be a clear voice for Australian Amiga users.

If you would like to see your artwork on the Portfolio page just send it in. All artists whos work is displayed will receive a FREE 3 edition subscription.

Still the only Australian Magazine with a Coverdisk.

Back issues are available in limited quantities. Please send $5 for each issue plus $1.50 postage.

Output is handled by:

-: The TypeHouse

Printing by:

-: Mercury Press

Distribution by:

-: Newagents Direct Distribution

Triple "D" Publications
PO Box 567
Mirrabooka
WA 6061

OZAmiga Magazine

This months cover pic.

Supplied by Colour Computer Systems. It was drawn using Real 3D version 2 and is one of the pictures included in the package.

THE CREW

David Reeves	EDITOR/ ADVERTISING MANAGER
Chris Leathley	PROGRAMMING EDITOR
Neil McKnight	AMOS EDITOR
David Perkovic	AMIGABASIC
Mark Little	C TUTORIALS
Juan Boulter	COVERDISK ADVISOR
Raff Lerro	TECHNICAL CONSULTANT

CONTRIBUTING AUTHORS
Bill Holder
Ian Harris
Mark Johnson
Nic Wilson

Disclaimer

The OZAmiga is an entirely independent publication striving to be a clear voice for the many talented Amiga users throughout Australia. The publishers do not accept any liability for any incorrect statements or information found in articles contributed by independent authors. The OZAmiga magazine in no way condones the illegal duplication of commercial software.

Subscribe and Save

EDITORIAL

Here we are again with our latest mag of goodies, read on and have some fun.

In this edition there are a number of new columns and the old ones continue in their recent vein.

Chris Leathley exceeds his earlier standards as he starts on the COPPER, in his Assembly tutorial. He has included six specially written examples with the executable files as well, on this editions coverdisk.

Raff persists with his look at how we use our modems to get the best results. These articles have been very well recieved and seem a boon to many people.

Neil takes us through his methods within AMOS to construct a simple paint program. He has included both the source and the compiled version on the coverdisk to give extra help.

Ian Harris again risks much criticism by writing the Dear Denise letters. It seems there are many people of different opinions regarding Dear Denise, but for now she stays!

We have also been joined by two new tutorial writers. The first being Mark Little from South Australia, who has begun to look at "C" and the best way to utilise it. The second being, David Percovic, also from South Australia. David has started his look at AmigaBasic in this edition. Many say it of no use going on with AmigaBasic because it is not supplied with the Amiga any more but there are many good after market Basic programs around with very few differences.

Also in this edition Nic Wilson of NO-VIRUS fame gives his unbiased look at the A4000, it won't please everyone but then it was't meant to. It was meant to give the honest truth as seen by Nic Wilson.

We look at Media-Flex, a "WOW" system designed for multimedia, Real 3D version 2, Virtual Reality and much more.

Thats enough from me read on and find out what every one else has to say.

Dave.

Letters to the EDITOR

ED.

Dear David,

I must applaud your publication, it's the first one that is 100% AUSSIE and has a useful coverdisk.

Your Vol 1 Edition 5 is great compared to others without naming names. OZAmiga has less B/S like pages of adds and useless reviews on other machines.

The main pee off is, the coverdisks the others try to provide tend to contain viruses, aren't configured to run on all workbench versions and they include games which look great on the cover but have hopeless game play.

To think that I pay 10 - 15 bucks per mag is very disappointing. I hope your mag can rectify these problems.

On a good note, I have found your mag very helpful with AMOS and was most impressed with the articles on modems as I have been thinking of getting one.

Congratulations to Raff for his inexpensive technical fix to an expensive problem. Speaking of problems, why would my stereo monitor be making a high pitched squeal that only stops temporarily when I give it a good nudge?

Your loyal supporter,
Richard King
NSW

Dear Richard,

I would like to say how happy I was to receive your letter, the praise I receive from people such as yourself is my inspiration to continue to make the OZAmiga better, whilst keeping the content not only of interest to people but useful as well.

With regards to the three points you raised concerning cover disks I must say that viruses are of paramount importance to all Amiga uses. If you come across a virus that your virus utility cannot kill (god forbid), I urge you to send it to one of the very talented people around the country that create the virus killers.

By doing this they can find a way to kill it and include it in their brain file updates. This helps to stop the spread as quickly as possible. Our concern about viruses is the reason why we include a virus utility or brain file on most of our coverdisks.

We most certainly wish to try to make our coverdisk compatable with all versions of workbench but some of the utilities we include are written for specific versions, sorry about that.

As for games on the coverdisk, one thing we will not do is load the disk up with useless games. Our coverdisk will only ever carry games written in Australia but mostly we would like to give you programs that may help you to become a more productive Amiga user.

Lastly, the problem with your monitor sounds very much like there is a dry solder joint somewhere inside. You can either try to find it yourself (only if you know what you are doing) or you can take it to almost any electronics workshop, explain the problem to them and they should fix it for a small charge.

I hope this helps, bye for now,

Dave

Dear Dave,

One aspect of computer useage that could do with a little bit of additional support is programming. Magazines do produce helpful and educative articles for programmers in the form of tutorials and reviews of various software authoring packages. However there is another angle which has distinct possibilities, I feel, for providing a welcome service to those of us who engage in the dubious and often frustrating art of creating new applications.

In similar vein to user group and BBS information, why not develop and publish (say quarterly) a list of people who dabble in programming for the Amiga.

Such a list could be compiled as the result of an invitation from OZAmiga to those kindred spirits who would welcome the opportunity to extend their programming contacts and thus benefit from the mutual help and sharing of ideas which should result.

The invitation could perhaps ask for such details as:

* Name
* Address
* Phone number
* Languages Used
* Programming interests
* Contact preferences

There must be quite a few Amigans of varying programming skill levels (beginners to experts) who have to "scratch their heads for solutions" at times alone, simply because they don't have any or enough avenues of assistance.

I hope you agree that this is a virtually untapped area of possible service and that anything that encourages and helps people to make usage of the Amiga more productive, more fun and more rewarding must be good for it's future.

OZAmiga is a most welcome, high standard addition to our computing world - keep up the excellent work!

Sincerely,
Trevor Parker
NSW

Dear Trevor,

Your suggestion was received with some shock as I had been approached by a personal friend regarding the same thing not three days prior to reading your letter. Needless to say when something is put forward with this much enthusiasum, I would be a fool not to take some note.

Therefore I will ask for programmers to send in this information (excluding Address) and publish it on a regular basis in a section called "Programming Pals". The reason for the exclusion of people's address is simple, we don't want equipment to start disappearing because we were advertising where it is (there are a few dishonest people around).

If the response is good enough (if we get many write in) we will see part 1 in edition 7.

Thanking you,

Dave

Dear Dave,

I too was going to write how ecstatic I am to have found a good locally produced Amiga magazine (if you can call 3500 Kilometers local?) but you must be getting bored of that by now, so I'll get straight to the point.

I was impressed by a couple of items in your PD pages, particularly "Phantasmos", which you said were available from Amilight in South Perth. Unfortunatly, the address was nowhere to be found.

Without pushing you into unpaid advertising, could you provide me with either details of Amilight or preferably a comparable PD library on my side of the country?

Also I have just written a music demo that I would like to release as public domain. I have enclosed a copy. How about a brief write up of a home grown demo in your PD pages? If you don't like it, feel free to say so (I won't be discouraged, I'll pretend you're just jealous). I think it is just as important to let people know what to avoid as it is to tell them what is brilliant in this field.

All the best,
Robin Davies
NSW

P.S. If worthy, could you forward my disk to Amilight or any other PD library you chose. Cheers

Dear Robin,

The fact is I never grow bored with hearing how good the mag is doing but it's OK if you don't wish to heap me with praise, I'll understand. I'm a big boy now.

Anyway It is good to hear from someone that takes an interest in PD Without giving away free adds to Brad, his number is (09) 367 4482. I have not really been given much information regarding PD libraries on the East Coast apart from the fact that most user groups hold respectable libraries or they can point you in the right direction.

With regards to your Music Demo I have given it a small mention on the PD page (hope you approve) and have also passed it on to Brad. It is always good to see people here producing work of this quality.
All the best.

Dave

Dear David,

Congratulations on having an Australian Amiga magazine, I hope you will continue to prosper and help to keep us all informed and up to date on the local Amiga scene.

I enjoy the informative articles and hope you will continue the assembler articles as I find writing assembler to be extraordinarily difficult on the Amiga, although I used to be quite a whizz on my Z80 based micro.

I hope that as soon as you have enough material to fill the mag, that you will drop the "Dear Denise" letters which seem to be aimed at a very low cultural level, perhaps I dislike it because I am no longer a teenager. I believe the column tends to denigrate the majority of Amiga owners/users.

I have a few questions which you may be able to answer. I produce a couple of newsletters here and am about to pension off my old, but trusty Brother HR15 Daisywheel for something faster and more versatile. The problem lies in finding both a printer and a word-processing program which will give me the greatest versatility and closest possible output to typesetting quality, at a realistic price.

I particularly need a nice, proportionally spaced, typeface which can be produced in justified mode. I have tried Word Perfect but am unable to install the typefaces I use on the Brother printer.
Scribble - Platinum Edition, which I am currently using, cannot cope with the proportional spaced copy and has several sever bugs, especially when changing measure which leaves odd lines incomplete.
Of course Scribble does have a good spell checker and thesaurus.

My second question is, what should I do with my old Amiga 1000 and sidecar? I would prefer that this not be answered by Denise who I am sure can think of the obvious answer after reading my second paragraph!

Finally, even though I am in the electronics industry I am very confused over the memory chips needed to expand my Amiga 2000HD. The chips listed in the handbook do not seem to be available here. Are ANY 254K x 4 bit chips with a speed of 10ns or faster OK?

With best wishes for the future of your magazine,
Colin Tringham
NSW

Dear Colin,

Thank you very much for your letter, it is good to get conflicting opinions regarding the articles we publish. In this way we decide what is to continue and what is to be junked. Unfortunately (for you) the number of letters that support the comedy in the Dear Denise letters far exceeds (for the moment) those that feel as you do.

Regarding your printer and wordprocessor, I can only suggest a Panasonic KXP2123, this is a 24 pin dot matrix that can easily be converted to colour. If this is a little out of your price range then you may like to look at a 9 pin printer. You must realise that these are only suggestions and that the best thing to do is go to various stores and look at what is available. For the wordprocessor I suggest Final Copy V2 or a desk top publisher like Pagestream or Pro Page. These will give a much more professional result that the old style wordprocessor.

Your Amiga 1000HD I'm afraid doesn't look good. You could try to sell it (unlikely at this stage) or perhaps donate it to a user group. Other than that you may try to set it up as a controller for all of your electrical products and gizmos. Unfortunately there is not much else you can do with it that I know of.

Lastly, the easiest way to find out which chips you can use, is to get a technician to look at your machine (it is very difficult to know which chips to use unless you have the machine configuration) and they will tell you. I hope I have been of some help to you although I think Denise could have answered you question regarding the A1000 with a bit more flair.
Regards
Dave

Programming in
ASSE
With Chris Leathley

THE MEGA MIGHTY COPPER

"Baby you can drive my car", oops I forgot where I was, the McCartney concert is long gone now, still I enjoyed myself very much. All I want to know now is when are Pink Floyd coming over? Anyway back to reality.

The Copper is one of the more versatile co-processors the Amiga has for offer. The Copper in general controls all of the graphic output to the monitor. It allows the screen to be split up vertically into sections, to have different picture resolutions and colour palettes. The Copper can also control most of the other custom chips.

The Copper is called a co-processor because it, like a real processor, has a program which is stored in memory. It executes each Copper instruction one at a time.

Ok now a little theory. Your TV or monitor has a electron beam, which basically starts at the top of the screen and travels left to right drawing the display. When it gets to it's right most position it will jump down to the next left and quickly move back to the left. Same as when the beem gets to the bottom it will move back up to the top of the screen and start again. The areas where the electron beam is moving to it's new position is called the "Blanking Gap". All this happens 50 times a second. For our sake we will call this a FRAME. The computer tells the monitor when to start drawing a frame, this is called synchronization. The computer and the monitor need to be synchronized as the Copper is a real time device. It looks directly at the beam position and works on that.

As mentioned above the Copper has it's own instruction set, but don't get too excited as the Copper has only 3 instructions. WAIT, MOVE and SKIP. I've never found a use for the SKIP instruction or seen it used in any copperlist I've come across, so it won't be covered. The two

instructions *The WAIT* and *MOVE* command may sound limited but they are in fact very versatile.

* *WAIT* for a specific screen position given as x, y coordinates.

* *MOVE* the immediate data into one of the special purpose registers.

We've talked about the Amiga's custom chips before now. In the Amiga's memory there is an area between $00DFF000 and $00DFFFFE that the custom chips use for their registers. These registers (addresses) contain information vital to the Amiga actually running at a hardware level. Because of this some of the registers are READ-ONLY and can't be accessed by the 68000. Other addresses are WRITE-ONLY and some are classified as STROBE, which mean it doesn't care what is written there, it only has to be toggled (like a switch (ON or OFF)). All of these registers control the Copper, Blitter, bitplanes, audio, colour RGB values, sprites, disk i/o, mouse/joystick inputs, serial control, interrupt control plus all of the beem control hardware.

ALL hardware registers have an EVEN address (divisible by 2). Usually the operating system takes care of this for us, but since we all want to know how to program the hardware directly, we will be covering all aspects of this as we progress.

The program that the Copper executes is called a COPPERLIST. It must reside in CHIP memory for it to work. Data for ALL custom chips MUST be in CHIP. The Copperlist is executed once per frame (50 times a second). It is automatically triggered at the start of the vertical blank. A Copperlist instruction is made out of 2 data words or 1 long word. It checks to see if bit 0 is set in the first word and decides if it is a WAIT or MOVE instruction.

MBLY

The *MOVE* instruction

The MOVE command transfers data from RAM to one of the hardware registers. The data to be transferred is the second word of the instruction. The register to be written to is the actual MOVE command. As stated previously, the hardware registers are situated on even addresses (they won't have bit 0 set), so when the Copper checks bit 0 in the first word it will automatically know if it is a MOVE instruction.

In our examples this issue, we will be only using one hardware address ($00DFF180). This address is the register for the background colour or colour 0. All colours on the Amiga are made out of a R (red), G (green) and B (blue) value. Each individual colour value has a range of 0 to 15 ($0 - $F hex). The higher the number the brighter the luminosity, a value of $F00 would be bright red while $800 will be half as bright. $FFF is pure white and $000 is pitch black. A play around with a colour slider program (found in Dpaint or Preferences) will allow you to mix various levels of each base colour to make lots of pretty colours. OK art lesson is over.

The Copperlist being word based cannot access addresses as high as $00DFF180 so it always adds the hardware base address ($00DFF000) to the move command, so a Copperlist command like this.

COPPERLIST: dc.w $0180,$0FFF

would add $00DFF000 to $0180 to come up with $00DFF180 and then move $0FFF into that address, thus turning the screen white. The Copper is a very fast processor and will take very little time away from the 68000 and Blitter. The bigger the Copperlist the more time it will take up but even that is not worth bothering about.

The *WAIT* Instruction

The WAIT command causes the Copper to wait until the video (electron) beam counters are equal to (or greater than) than the coordinates supplied in the instruction. While waiting, the copper doesn't affect the 68000 or Blitter. The first instruction word contains the vertical and horizontal coordinates to wait for. The second word contains information on how to do the beam position comparison. (I will not be going into this at this time) We will be using $FFFE in this field.

VERTICAL BEAM POSITION

The vertical beam position is given as the first byte in the first word. Since we can only use a byte for the value, it can only be in the range of $00 - $FF. This is the standard range for a NTSC (American) machine, but for us PAL users we want to be able to use the bottom 2 inches of the display. We use a special WAIT command to trick the copper into missing the first VERTICAL BLANK position. It will however happen about 56 beam lines after that. I'll cover that special WAIT position a bit further down.

HORIZONTAL BEAM POSITION

The horizontal beam position takes up the second byte in the first word and can have a value between $00 and $E1. It must be an odd number so the copper can tell that it is a WAIT instruction (bit 0 is set). The Copper has a horizontal waiting resolution of 4 lo-res pixels or 8 hi-res pixels, so there is only 113 positions available for the copper. In general programming the horizontal position isn't used, so unless you want to do some fiddly screen manipulation you don't have to bother with it.

The WAIT instruction can look something like this.

COPPERLIST: dc.w $9601,$FFFE

dc.w $FFFF,$FFFE

This WAIT instruction is used to tell the Copper to stop and wait for the next VERTICAL BLANK.

A very small Copperlist program to tell the display to change the background colour on line $80 (128) to a bright blue.

```
COPPERLIST: dc.w  $8009,$FFFE
            ; wait for line $80
            dc.w  $0180,$000F
            ; set to blue
            dc.w $FFFF,$FFFE
            ; stop Copperlist
```

As mentioned previously the Copper can only wait for a scan line between $00 and $FF. We use this wait instruction $FFDF,$FFFE, to trick the Copper into going into the PAL section of the screen. The beam position is set back to 0, so to wait for line $120 we use just the low byte $20. This Copperlist example will set the screen to be yellow in the standard PAL picture range (just like your WorkBench).

```
COPPERLIST: dc.w $2C09,$FFFE
            ; $2C is the top of
            picture
            dc.w  $0180,$0FF0
            ; screen now yellow

            dc.w $FFDF,$FFFE
            ; miss vertical blank

            dc.w $2C09,$FFFE
            ; $2C ($12C) is
            bottom of picture
            dc.w  $0180,$0000
            ; screen now black
            (kill yellow)

            dc.w $FFFF,$FFFE
            ; stop copperlist
```

Your standard workbench screen is 256 pixels high or $100 hex. It will usually start at scan line $2C (44) and end on $12C (300). Bitplanes will be covered in the next issue of OZAmiga.

OPERATING THE COPPER

A Copperlist is not sufficient to operate the Copper. We need to be able to tell the Copper where the Copperlist is. The Amiga's hardware has two Copperlist pointers (WRITE-ONLY). The reason it has two is that interlaced pictures require two Copperlist programs. One Copperlist displays all the even lines while the other displays odd lines. These registers are called:

COP1LC ($00DFF080)
COP2LC ($00DFF084)

COP1LC is triggered by the vertical blank so we will be using this register.

We also need to turn on the Copper DMA so the Copper will work, otherwise it won't have any hands to grab the instruction from the Copperlist. The DMA control registers DMACONW ($00DFF096 - WRITE ONLY) and DMACONR ($00DFF002 - READ ONLY) instruct the Amiga what DMA channels to use. Here is a run down of what they are.

BIT	NAME	FUNCTION WHEN SET
15	SET/CLR	Set or clear bits
14	BBUSY	Blitter busy (read only)
13	BZERO	Result of blitter operation is 0 (read only)
12	UNUSED	
11	UNUSED	
10	BLTPRI	Blitter has priority over processor
9	DMAEN	Enable DMA set below (main switch)
8	BPLEN	Enable bit plane DMA
7	COPEN	Enable copper DMA
6	BLTEN	Enable blitter DMA
5	SPREN	Enable sprite DMA
4	DSKEN	Enable disk DMA
3-0	AUDxEN	Enable audio DMA for sound channel x

The DMACONW register is not written like a normal register. Bits can only be set or cleared. This is determined by bit 15 in the data word. If this bit is set, all set bits of the data word are also set in DMACONW. If bit 15 is clear then only the set bits in the data word are cleared in DMACONW. All other bits remain unchanged.

Bit 9 is something like a main switch, this must be set for any DMA to work. The only DMA to be used is that which have been set to 1.

NOTE:
When running stuff from the operating system most of these bits will already be set. Don't experiment with bits that you don't know about as it can cause some chaos.

To turn on the Copper we move $8280 (bits 15, 9 and 7) into DMACONW. The copper will then fire up and run the Copperlist.

Included on the coverdisk are 5 examples of what can be done with a Copperlist and the source to them. The first example "Copper1" will put 7 rainbow coloured bars onto the screen. This first program is very well commented and will explain at a 68000 level what instructions are needed. The second example "Copper2", shows you how the Copperlist can split up the display in the horizontal axis and how fast the Copper works.

Most programs that use the Copperlist (ie/ games, demos) are all timed to the vertical blank, or a particular beem scan line. This allows the program to run at a set speed. The Amiga keeps a counter of what scan line it is currently on in the VHPOSR register ($00DFF006 READ-ONLY). We only need to check the first byte of this address (the vertical position), as the 68000 isn't really fast enough to get a match on the horizontal beem.

```
WAIT_VB:cmp.b #$FF,$00DFF006
bne  WAIT_VB
```

Those instructions wait for the VERTICAL BLANK or scan line $FF. It is usually the first set of instructions in the MAIN LOOP of a program. We must be careful of where we wait for, as we cannot modify any DMA registers while they are currently being used. That's why the VERTICAL BLANK is recommended or an area where you know what is happening (or not happening in this case)

"Copper3" is just a little program that uses the 68000 to modify the Copperlist to move a white line up and down the screen.
"Copper4", scrolls the colours in a section of the Copperlist around on it's self.
"Copper5", cycles a rainbow colour list though two types of displays. The first is a single Copper line split into 50 horizontal sections while the other is a Copper block made up out of 32 lines.

Go through the example programs carefully and try to understand what is going on and how the 68000 routines work. Modify them if you like and definitely try to create your own Copperlist. Start nice and simple and then work your way up. Don't panic when if the machine starts to freak out, it's all part of the learning process. The Copper is a very powerful chip, and when used correctly it can create very stunning displays on the screen. All of the Example programs only work on 1 colour register, imagine what you can do with the rest. They will be delt with next time in bitplanes.

Before I forget, last issue Raff mentioned a public domain program to toggle the lo-pass filter on and off called LED. Well also in the assembler subdirectory is a two liner called LED (funny that). It will toggle the filter on or off every time it is called. The source is also included for those who want to know how it's done.

Have fun with the COPPER, see ya next time.

ZAPHOD

Send enquiries to:

Assembly Questions
PO Box 567
Mirrabooka
WA 6061

Modems for the BBS User

By Raff Lero

PROTOCOLS

Hi, and welcome to part three in this episode, I hope to explain some of the mysteries of transfer protocols, and hopefully explain which protocol is best for you.

Firstly, to recap, *what is a protocol?* Well, in our case a protocol, is the agreed method and set of standards, that we are going to use to perform our file or text transfers between computers. Fortunately, unlike many other things, there are several common standard transfer protocols the world over, great hey?

In the early years of computer to computer communications, many different people had many different ideas about just how things like files and text should be transferred and of course there were many oddball methods about, most incompatible with each other. Luckily, these days it has all settled down somewhat to just a few fairly standard methods.

A very old, and seldom used protocol is X-Modem, when I say "seldom used" I mean, with regard to the Amiga. Windows 3.1 has a terminal program included which has this and another rarely used protocol "Kermit", but that is another story. Kind of says something about Windows maybe? Anyway, X-Modem, although used infrequently, is still useful if communicating to older machines or older programs unable to support the newer and better protocols.

X-Modem has only a few options and switches available to the user:

(T) Text mode on or off, off being used to transfer binary or program files
(C) CRC checksum on or off, on being to enable crc checking of transferred data. (See my previous article "Part Two" for data about CRC checking)
(K) Block size 0=128 byte blocks, 1= 1k byte blocks. This switch adjusts the size of transmitted/received blocks of data. 1k blocks are preferable because they create less overheads, and therefore faster transfers, 128 byte blocks are sometimes useful with very noisy lines.

That is it for X-Modem's switches.

X-Modem is a protocol which sends either 128 byte or 1k byte blocks as discussed earlier and after each block expects an ACK character from the receiving end to Acknowledge a good block, or a NAK character to Negatively Acknowledge a bad block to be resent.

Because of all of this ACK/NAK stuff going on and the fact that as standard it only sends 128 byte blocks it is quite slow.

X-Modem is also unable to send either the file name of the transmitted file, or multiple files in a batch. This means the user must name the file being transmitted, and also monitor the transmission if more than one file is to be transmitted.

Enter Y-Modem.

Y-Modem is one of the first protocols to support batch files, *hooray!!*
Y-Modem is also apparently compatible with X-Modem.

The switches for the version of Y-Modem that I have are:

X- /Y-Modem, enables X-Modem emulation.

Y-Modem batch, enables batch file modem, ON by default.

Y-Modem -G, uses 1k byte blocks and a 16 bit CRC for better throughput.

Y-Modem -A, for even better throughput, uses 2k byte blocks but NO error checking. OK if you are using a modem in RELIABLE mode. (See Part two of this series for information about Reliable mode modems).

NOTE:
Y-Modem -G and -A work the same as Y-Modem batch, but they don't send an ACK character for each block nor can they handle CRC checking or errors.

Y-Modem has some other options as well:

Chop option "Z" will chop files only if the file size is unknown. Enabled by default.

Output option Ö" this specifies the output directory for received files, by default it is RAM: on the Amiga.

CRC option "C" when enabled, Y-Modem will use 16 bit CRC error checking, or if not enabled it will use the checksum method enabled by default.

1K byte blocks "B" when enabled Y-Modem will use 1K byte blocks, when not enabled, it will use 128 byte blocks enabled by default.

These options will either be supported by your terminal program, or can be entered with commas between them. Eg. Z0,C1,B1 = Chop disabled, CRC on, and 1K byte blocks.

Next comes Z-Modem.

Z-Modem has a multitude of options and switches making it very versatile. It also supports batch file transfers, resume mode (where a previously uncompleted transfer may be resumed where it left off, useful for disconnection's etc). It also has an automatic challenge mode among other things.

T{Y| N| ?| C}
Text translation mode:

TY = Text Yes; if receiving, translate CR/LF pairs or solo CR chars to normal Amiga LFchars. If sending, suggests to receiver that they should receive this file in text mode.

TN = Text No: receive file verbatim, without changes. If sending, suggest to receiver that they receive this file verbatim, without translations.

T? = Text status unknown; if receiving, use sender's

suggestion as to whether to do EOL translations or not. If sending, tell receiver to use default mode, because we don't know either.

TC = Text mode set by Communications program; the protocol asks the communications program whether or not to use Text mode for each file. If the communications program doesn't support the necessary xpr_finfo() call, or if the call fails, this option acts like T?.

From the user's point of view, what this option normally does is set the Text mode to match the communications program's built-in text/binary/end-of-line/translation mode, if any.

O{Y| N| R| S}
Overwrite mode:

OY = Overwrite Yes; if about to receive file with same name as one which already exists, delete the old file and receive the new file in its place.

ON = Overwrite No; if about to receive file with same name as one which already exists, append ".dup" onto the name of the new file to keep them separate.

OR = Overwrite Resume; if about to receive file with same name as one which already exists, resume receiving file data from the current end of the existing file.

OS = Overwrite Skip; if (etc.), tell sender never mind, skip this file, we don't want it. Batch transfers will move on to the next file in the set, if any.

Bnnn Buffer size:

Z-Modem adds a layer of file I/O buffering in addition to whatever the comm program may or may not provide. This option sets the size of Z-Modem's file I/O buffer in kilobytes. The minimum value is 1 KB, for those using RAM drives or fast hard drives, or those whose communication programs already provide sufficient buffering. The maximum value is as much continuous RAM as you have available in your Amiga. If you specify more than is actually available, Z-Modem will keep decrementing the buffer size requested by 1 KB until the memory allocation works. That way, if your RAM is too fragmented to use the amount you request, Z-Modem simply uses the

largest block available. Buffering is especially helpful for floppy drive users; it keeps your drive from continuously gronking and slowing things down all through the transfer.

Fnnn Frame size:

Although normally avoided, Z-Modem has the ability to require an ACK to be sent from the receiver to the sender every X-many data bytes. Normally you don't want to use this feature, because not waiting for ACKs is part of how Z-Modem works so fast. However, this feature can be very useful in conjunction with file I/O buffering on slow devices (namely those floppy drives). If you set up a large I/O buffer to avoid gronking your floppy so often, you'll find that when the buffer finally does get around to being flushed that it can take a long time; so long in fact, that the delay can cause timeouts and errors. But if you set your Z-Modem to require the sender to wait for an ACK every buffer's-worth of data, the sender will politely wait for you to flush your buffer to the slow floppy and send it an ACK saying it's OK to continue now. This value should be set to 0 to disable ACKs (normal mode), or set it to the actual number of data bytes allowed between ACKs. For example, if you set B64 because of your floppy, you should also set F65536.

Ennn Error count:

This allows you to set the number of sequential errors which will be required to convince Z-Modem to abort the transfer. The normal value is 10, meaning that 10 errors must happen in a row with no valid data being transferred in order to cause an abort. This setting is provided for those using Z-Modem with a BBS, who may wish to use a relaxed setting, or those with really lousy phone lines who are desperate and patient enough to want the transfer to continue in spite of horrible performance.

A{Y| N}
Auto-activate mode:

AY = Auto-activate Yes; if your communications program supports the ability, Z-Modem will automatically go into receive mode when the start of a Z-Modem download is detected.
AN = Auto-activate No; don't try to automatically start downloading, make the user activate it.

D{Y| N}
Delete after sending:

DY = Delete Yes; delete each file after it has been successfully sent.

DN = Delete No; don't delete files after sending them.

K{Y| N}
Keep partial files:

KY = Keep Yes; keep the fragment of a file received so far if file reception is aborted. This allows you to use the Overwrite Resume option above to pick up where you left off on your next attempt.

KN = Keep No; delete any partially-received file after an aborted transfer.

S{Y| N}
Send full directory path:

SY = Send path Yes; send full filenames including directory path to receiver.

SN = Send path No; send only simple filenames, not including directory path.

R{Y| N}
Receive full directory path:

RY = Receive path Yes; use full filename exactly as received, instead of using the P option directory path.

RN = Receive path No; ignore received directory path (if any), use P option directory path instead.

P{dir} Path to use for
received files:

Px = Store all received files in directory "x" if option RN set. Ignored if option RY set. "x" can be any valid existing directory, with or without trailing "/" (eg. "Pdf0:", "PComm:hold" etc.).

If setting the options via the option string method (either ENV: file or primitive comm program), note that the setting for each option must immediately follow the option character with no intervening characters ("TY", not "T Y" or "T=Y"). When setting multiple options at once, separate the options from each other with commas and/or spaces; for example, "TN,OR,F0". You don't have to specify all options every time; the options you specify

will be merged into the current option settings, replacing their old values. Upper/lower case is not significant. The default option settings if you don't change anything are "TC, ON, B16, F0, E10, AY, DN, KY, SN, RN, P".

Serial port settings:

Z-Modem requires that your serial port be set to 8 data bits, no parity, 1 stop bit. This allows Z-Modem to send full 8-bit binary data bytes without having them munged on the way through the modem. If your communications program supports the xpr_setserial() function, Z-Modem will use it to set your serial port to 8N1 before doing a transfer, and will set your port back the way it was again after it's done. If your communications program doesn't support xpr_setserial(), you'll need to make sure it's in 8N1 mode yourself. Z-Modem works well in all serial port hand shaking modes; none, XON/XOFF, or 7-wire/RTS/CTS. Since any or all of those hand shaking modes may be appropriate at different times, with different modems or remote systems, Z-Modem lets you set the hand shaking mode and doesn't mess with it.

Receiving files with Z-Modem:

Once you get the Z-Modem options and your serial port configuration set up properly, you're ready to actually use this thing (gasp!). Receiving files via Z-Modem is quite simple. First, get the file sender going by using whatever command it wants.

Z-Modem is a batch file transfer protocol, meaning that it's capable of transferring several files in a single exchange, so the remote system may allow you to specify multiple files to be sent to you at one time. It may also allow you to use wildcard characters in the filename(s); this is all system dependant.

This may be all you have to do. If you specified option AY (auto-activate on), and your communication program supports it, Z-Modem should automatically activate at this point and start receiving your files. If you specified AN, or you communications program doesn't support auto-activation, you should now use whatever option your communication program provides to activate file reception; this will usually be a menu option or button gadget. Either way, once Z-Modem starts receiving files, it should automatically receive all of the files you specified into the proper

directory as indicated by the R and P options.

Make sure that you have set the Z-Modem options properly before starting the transfer; especially, make sure you only use TY if you know that all of the files being transferred in this batch are printable ASCII text files. If you use TY on normal binary files like .ARCs or .ZOOs, they'll be mangled beyond use.

Sending files with Z-Modem:

Sending files using Z-Modem is fairly straightforward. First, activate the file receiver with whatever command the remote system requires. You may or may not need to specify a filename or directory to the remote system; this depends on their implementation of Z-Modem. Once the remote system is ready to receive files, activate your communication program's Z-Modem send function. Your communication program will prompt you for which file(s) to send.

Although Z-Modem is a batch protocol, your communications program may or may not allow you to specify multiple file names to be sent; also, wildcards may or may not be supported. These decisions are up to the communication program's author; Z-Modem will handle multiple files and wildcards if the comm program allows them. Once you've specified the file name(s), the file(s) will be sent to the remote system. Note that the T option serves only as a suggestion to the receiving system when sending files; the receiver makes the final decision as to whether to take your advice or to force the files to be received in text or binary mode.

Normally, file transfer protocols have to compromise between efficient data transmission on good, clean phone lines and quick error recovery on bad, noisy phone lines. If you pick a large packet size, you get high throughput on clean lines due to low packet overhead, but you have slow recovery times and large amounts of retransmitted data on noisy lines. If you've used Y-Modem on noisy lines you've seen this problem. But, if you use small packets to reduce retransmitted data on noisy lines, you increase the amount of time the protocol spends sending packet overhead and your throughput suffers. The solution is to vary the block size according to the experienced error rate during the transfer. That way you aren't stuck with a rigid packet length which will sometimes be the wrong size no matter what.

These three protocols are by no means the only ones used, but they are certainly the most common.

You will hear about others such as Sealink, Kermit, J-Modem, and several others, but I won't go into these here, as they don't seem to be in common use in the Amiga community.

There is also a Bi-Modem protocol, which I believe has the ability to up and download simultaneously, but I have no documentation on it as yet, and only certain newer communications programs will be able to take advantage of it. Term for the Amiga using Workbench V2.x is a good example.

Next Issue, I hope to discuss Archival programs and there uses.

Until then,

B.C.N.U.

Raff....

Mark Little

Joins us with the long awaited

Learn C

PART 1

The 'C' language is a popular, medium level language developed by Bell Laboratories to write the UNIX operating system. It's medium level because it deals with data items such as floating point numbers, strings and arrays, but cannot handle them as a single entity. For example, you cannot copy a string by saying "A := B" as you can in Pascal.

The best way to understand a language is to use it, so let's look at a few simple programs. In the "Learn_C" directory on your disk is a source file called "Hello.c" which displays a message. There are many comments in the source file, but it is similar to the listing below.

```
/* This is a comment - see how it
starts and ends */
#include <stdio.h>
(void main(void)
printf("Hello World - Nearly
everyone's first program\n");}
```

The first line is a comment. A comment starts with "/*" and ends with "*/". It can take as many lines as you like.

The second line reads from a file which defines the standard input and output functions because the output function "printf" is not part of 'C'. The angle brackets around the file name mean that the compiler should look along the search path for the file, while double quotes (") mean it should only use the current directory.

The "include" files list functions which are available. When you compile your program, it records which external functions it needs to work. In most compilers, the "include" files are just text, so you can see what they define. Joining external functions to your program is a function of the linker and your compiler's instructions will show how to do this.

The third line tells the compiler where to start the program. The first "void" means that the "main" procedure will not return a

result. The second "void" means that we don't need any data to be supplied. Later programs show how you can get the command line by replacing the second "void" with other arguments. The curly braces "{ }" show what instructions belong to the "main" procedure.

The next line displays the text inside the brackets. Running the program, will show you that it doesn't print "\n" after the word "program." In fact, it goes to a new line. The "\" character is special and shows that the next character is a format character. As you have might of guessed, the "n" stands for New Line.

If you want to use the backslash character in a string, you must use two backslash characters in a row ("\\"). You don't need to do this when your program is running.

At the end of the "printf" statement there is a semi-colon (;) which defines the statement end. Because the semi-colon marks the end of a statement, you can use as many lines as you need for a statement.

Using the documentation supplied with your compiler, compile the program and then link it to create a program (if you don't have a compiler, I suggest the public domain version of DICE (Dillon's Integrated C Environment). The sequence of compiling and linking is known as "building". A bit of computer jargon which speaks for itself.

DICE is available from most good PD libraries. If you cannot get it write to Mark at the supplied address and we will send it to you.

When you are happy building programs, try altering "Hello.c". Besides changing the words, you should try some special sequences such as "\f", "\t" and "\r". If you can't figure out what happens, I'll let you know in the next article.

As you can see, it is easy to send text to the display and what we have done is fine for headings. Displaying results is another matter as the output text must change as the results change. The "printf" statement allows this by using more formatting characters.

The example program "scanf.c" allows you to read in data from the console and use it in your programs. It also shows you how to use "printf" to change the text displayed. More importantly, it introduces the idea of variables and how to address them.

To explain addressing in every day terms, think about a house. A house has an address in the street and the house can contain people. A variable is like the house. You have to know what it's address is to be able to find out who lives there. You can find out who lives in the house by looking up the address in the phone book and then

looking in the house. In 'C', adding "&" in front of the variable gives you it's address. So if we say "A = &House;", the value of A is set to the address of the "House" variable.

Knowing the address, I can say, "Who is at that address?" In 'C' I would use "Who = *A" which means what is the content of the variable whose address is in A.

So, tying this together we have:

    A   = &House;
    /* Get House's Address   */
    Who = *A;
    /* Who is in the address */

which is equivalent to

    Who = House;
    /* Who is in the house */

This concept of pointers is hard to grasp, so don't worry if you have difficulty - I still get it wrong sometimes.

Data in brackets is what's passed to a function when called and always remains unchanged when the function ends. This is a problem when a function wants to modify the value(s). If we have a command

    SetToOne(A);

where it's to set A to 1, then we are trying to break our rule that values passed to a function are always unchanged when it ends. "A" will always be the same value as when it started. This is where the address of the variable comes in. Let's pass the address of the variable instead.

    SetToOne(&A);

We can now use that address to change its contents. Because the address does not change, we haven't broken any rules. What statement do you think we would use to change the contents of "A" using its address? If you think about the house analogy, you may be able to figure it out.

Well after that long winded discussion, let's get back to the new command

    scanf("%d",&Minutes);

This command says scan the keyboard until the user presses return (end of line) and try to make an integer number ("%d") out of the text. If you can make a number put it into the variable whose address is given (&Minutes). The next article discusses, what exactly an integer is, but for now, it's just a number.
There are too many format options to list here and I suggest you consult a book on 'C' for details. The sample

programs contain some extra examples.

The "printf" command sends the results to the display. As you may notice, there are "%" formatting characters in the text of the "printf" statement. These formats are very similar to the "scanf" formats.

Placing numbers between the "%" and the "d" sets the minimum number of spaces left for the number. It also sets whether leading zeros are made into spaces. The format string "%4d" leaves at least four characters for the number. (any leading zeros are represented as spaces) The format string "%04d", leaves the minimum number of characters at four, but prints any leading zeros.

The next article discusses data types such as int, byte, word, long and structures. Have fun and 'C' you next time!

Mark.......

P.S. To change the contents of a variable using its address, the statement would be

    *A = 1;

which translates to; Set the variable pointed to by the address in A to the value 1.

How to do
ASSIGNS

To start this article we must understand what an Assign is and what it does.

When you copy a program to your hard disk and try to run it, on many occasions you will find it asks you to 'Insert disk 2 in any drive'. The reason for this is that when the program was written it was told that it could find the information it is looking for on disk 2. Even though the data is right there where you put it on the hard disk the *£"!\^% computer keeps asking for disk 2.

Until you tell the computer that the information it is looking for has been moved to a new location on the hard disk, it will keep asking for the floppy. The act of telling this to the computer is called (as you may have guessed) ASSIGNing a disk.

To do the assign you must type the following line into the CLI.

ASSIGN <disk name>: dh0:<path to files>

If you intend to keep the program on your hard disk you should include the assign into the STARTUP-SEQUENCE (SUS) so that it is done automatically when you boot the computer.

The SUS is a small file that sets up the computer and loads the Workbench when you boot. It is located in the 'S' directory, usually with a couple of variations. To put the assign into the startup you need to load the SUS file into a text editor or word processor and add the line above.

You will find other assigns in the SUS, so enter the new one into the same area. Lastly, when you save your altered startup make sure it is saved as an ASCII file that overwrites the original SUS (you may find it prudent to make a backup of the original in case of problems).

If you have a particular area of trouble with your Amiga or it's set up, then write into us here and we will try to help you and those with the same problems.

Send enquiries to:

    Amiga Help
    PO Box 567
    Mirrabooka
    WA 6061

AMOS

With Neil McKnight

Graphics Part 3

This month we continue our tutorial on AMOS graphics, finally making something useful of them all by putting them all together to create a small paint program written entirely in AMOS.

We've been using various graphics commands for some time now. We have looked at drawing, changing colours, making patterns and even saving the results as a standard IFF picture. Now we are going to learn how to customise the display to get exactly the right resolution and number of colours we want.

What is Resolution?

Resolution is how many pixels we can fit into a given space. The more pixels, the higher the resolution. On the Amiga, we usually talk of High resolution being 640 pixels across the entire screen and Low resolution being half that number, or 320 pixels across the screen. If you prefer to think in terms of text, the normal Workbench screen is a high resolution one because you can fit 80 characters on one line.

AMOS normally creates a Low resolution screen for you when it first starts up. These screens can display only 40 characters on a single line. In both High and Low resolution screens the number of lines of text displayed is the same, just the width of the characters is different. You can also change the number of pixels that fit on a screen vertically. Normally this is 256 for a full sized screen in both High and Low resolution. By Interlacing the display, you can double this to 512 pixels or about 64 lines of text. You can tell if a screen is interlaced if the text looks squashed up or the screen flickers.

Screen Modes

The mode of a screen is a combination of its resolution and the number of colours you can have. A Low resolution screen on a

standard Amiga can have 2, 4, 6, 16, 32, 64 or 4096 colours. A High resolution screen may only have up to 16 colours, but the graphics are much finer. The newer AGA machines can have more colours, and have other screen modes too, but at present AMOS doesn't know about them. As I mentioned before, the Amiga only has 32 colour registers (pens), so asking for more than 32 colours forces the computer into special graphics modes called Extra-Halfbright (64 colours) and HAM (4096 colours). These special graphics modes will be covered at a later time.

Opening a Custom Screen

OK, knowing a little about screen resolution and having decided how many colours you will need, we can open our very own screen exactly how we want it.

Our first new command for this month is Screen Open, and as you might guess, it opens a custom screen for us.

> **SCREEN OPEN n,w,h,c,mode**

As you can have open up to 7 different screens simultaeously, we give each screen a number <n>. Supply the width we want (in pixels) as <w>, the height we want in <h>, and the number of colours in <c>.
For <mode> we can use either Lowres or Hires (just type the word). You can also turn on interlacing if you wish by adding "+Laced" on the end.

Examples:

> Screen Open 0,640,256,4,Hires
> just like the Workbench.
>
> Screen Open 1,320,256,32,Lowres
> like Dpaint Lowres.
>
> Screen Open 2,640,512,8,Hires+Laced
> 8 colour Dpaint Hi-res.
>
> Screen Open 3,320,100,2,Lowres
> a short screen.

> Screen Open 4,100,200,2,Hires+Laced
> Screen Open 4,100,200,2,Hires+Laced
> a tiny interlaced screen.
>
> Screen Open 5,1000,1000,32,Lowres
> a huge screen!

As you can see, you can specify almost any size screen you want. The catch is that each screen consumes precious Graphics Memory. If you're lucky enough to own a machine with lots of "Chip Ram", then its not such a problem. As a general rule, a Lowres 16 colour screen (320 x 256) uses about 41K of chip ram. A Hires screen will use twice as much, and interlacing it will double it again, to about 164K! By the time you take into account any graphics, sound or music you might want, all of which must be stored in chip memory, you begin to see why games do not use high resolution screens.

About Coordinates

In all the graphics commands so far, we've talked about points, pixels, widths, heights and resolutions. All these are based on various points on the screen. We call these measurements Screen Coordinates. That is, where a certain coordinate appears on the screen, depends on the type of screen it is on. But what do you use when you want to change to a position of the screen itself? As you might well guess, it is the Amigas hardware that controls all those screens. So we use another type of coordinates to tell the hardware where we want things. These are called Hardware Coordinates. The main difference between screen coordinates and hardware coordinates is that screen coordinates are relative to the edges of the screen you are using, but hardware coordinates are always fixed (because they're part of the hardware).

Controlling Screens

When you make your new screen, especially if it is smaller or larger than the display, you may want some way of positioning it where you want it.

There are two commands to control a screen's position and the parts of it you can see.

The first is Screen Display. This command uses Hardware Coordinates to tell the Amiga where to put the screen. The visible range of these coordinates is between 112 - 448 horizontally and about 30 - 300 vertically.

The syntax for this command is:

SCREEN DISPLAY n,x,y,w,h

It is fairly simple to use. The value <n> is just the screen you want to control, the <x> value is the horizontal position on the display, <y> is the vertical position, and <w> and <h> are the width and height of the area you want visible. A typical screen is positioned at about 128,42.

Try these in Direct Mode:

```
Screen Display 0,100,,
the screen shifts left.

Screen Display 0,,100,
the screen moves up.

Screen Display 0,128,42,,
back to normal.

Screen Display 0,,,100,100
see just a small part of it
```

If you get stuck, just type DEFAULT and the screen will be restored to its original state for you.

Now that we can place any screen where we want, you might find that the display gets a little cluttered. So here are all the commands you need to get everything just the way you like:

SCREEN TO FRONT <n>

This moves the screen <n> in front of all others, or the current screen if <n> is omitted.

SCREEN TO BACK <n>

This moves screen <n> behind all others. The current screen will be moved if <n> is not supplied.

SCREEN HIDE <n>

This hides screen <n> so you can't see it.

SCREEN SHOW <n>

This makes screen <n> visible again.

SCREEN <n>

This makes screen number <n> the current screen. All text and graphics will now go to that screen until you say otherwise. The current screen may or may not be visible as you choose.

Buttons and Things

One of the favourite things people like to program in AMOS is buttons. Ignoring AMOS Pro for the moment, because it has a whole series of button interface commands, I'll just use those commands available to everyone.

What is a button? A button is just some defined area of the screen that the user can click on using the mouse. It may look like an Icon, as in Workbench, or some text, or even an animated little man. In AMOS, buttons are easy!

AMOS can detect when the mouse is clicked in a certain part of the screen using zones. You define the zone and AMOS will tell you when someone is pointing at it. Simple!

Before you start, tell AMOS how many zones you want. Use the following command:

RESERVE ZONE <n>

Just set the value for <n> to the number of zones you need. You only have to do this once in your program, somewhere near the beginning.

Next, set up a zone for AMOS to check:

SET ZONE <n>,<x1>,<y1> TO <x2>,<y2>

If you have been following these tutorials, you will notice this is just the coordinates of the area you want defined, with a number <n> so you can tell which one of them the mouse is pointing at. Make one of these, using different numbers for <n> of course, for each button you want.

Now all you have to do is find out when someone moves the mouse into one of your zones. The command to do that is:

n = Mouse Zone

Here, <n> is the number of the zone the mouse is in right now. The best use of this is in a loop. Type the following lines and RUN it. To exit this little program, press CTRL-C.

```
Reserve Zone 3
Set Zone 1,0,0 To 50,50
Set Zone 2,100,100 To 200,130
Set Zone 3,180,25 To 200,200
Do
Print Mouse Zone
Loop
```

If you ran the above little program you will have noticed that zones are invisible, it's up to you to put something there for the user to "click on". Secondly, zones can overlap, and where they do overlap, the LOWEST number is the one you get.

To make real working buttons all you have to do is add a box where the zone is, and exit the loop when the mouse is clicked.
The press of the mouse key is sensed by the following command:

<n> = Mouse Key

If <n> equals 0 then no key was pressed. If the left button is pressed then <n> = 1, for the right button <n> = 2.

We can modify our previous program by waiting for the user to click on one of the zone boxes like so:

```
Reserve Zone 2
Set Zone 1,100,100 To 150,110
Box 100,100 To 150,110
Set Zone 2,100,120 To 150,130
Box 100,120 To 150,130
Repeat
  K = Mouse Key
  N = Mouse Zone
Until K=1 And N>0
If N=1 Then Print "Top Box Selected"
If N=2 Then Print "Bottom Box Selected"
```

This program creates two zones, one above the other, and draws some boxes so you can see where they are. Then

Then the program loops until you click the LEFT mouse button while inside one of the boxes. A message confirms which box you selected. AMOS also has an easy way to create text buttons. We'll look at text buttons next time, when we cover the various text commands.

Drawing With The Mouse

Anyone who has used an Amiga for long will have plenty of experience drawing all sorts of flowing shapes using the mouse. But how can we do this in our own programs?

As we already have learned heaps of different drawing commands, the next thing we need is to find out exactly where on the screen the mouse is, and the rest should be easy. Well it is, except for a tiny little twist. The mouse pointer is controlled by a special piece of hardware inside the computer. Unfortunately it will only tell us where the mouse is in Hardware Coordinates. As you remember, we can reposition the screen at will, so we need to be able to convert those hardware coordinates into the right screen coordinates so we can draw our line in the right place. Thankfully, AMOS provides all these commands. So here the ones you will need:

```
x = X MOUSE
y = Y MOUSE
```

X Mouse gives you the horizontal position of the mouse in Hardware Coordinates, here, into the variable <x>, and Y Mouse gives the vertical.

```
x = X Screen(xb)
y = Y Screen(yh)
```

X Screen converts the Hardware Coordinate <xh> into the Screen Coordinate <x>, and Y Screen does the same for <yh>. Using them together in a shorthand way we get:

```
x = X Screen(X Mouse)
y = Y Screen(Y Mouse)
```

This gives us the position we need to draw using the mouse.
Type in this little program and RUN it:

```
Repeat
Draw To X Screen(X Mouse),Y
Screen(Y Mouse)
Wait Vbl
Until Mouse Key>0
```

This keeps drawing a flowing line until you press either mouse button. Note the extra command WAIT VBL. This is used to synchronize the mouse movements with the graphics display.

Remove the WAIT VBL and see what the different is.

OK, that's about all for this edition. Next time we have a final look at graphics (for a while at least) and discover all about colour effects, colour animation and image compression, and probably start on our next topic: Text and Fonts.

If anyone has any specific ideas on what other areas they want covered, or supply AMOS hints and tips, then write to me here at the address below.

OZAmiga AMOS Section
PO Box 567
Mirrabooka
WA 6061

PS: Don't forget about the OZAmiga AMOS competition. YOU could win for yourself a copy of the AMOS Professional valued at $229 rrp.
To qualify, send any ORIGINAL piece of software written using AMOS that will fit on a single floppy disk and you are willing to release as Public Domain (not Shareware).
Entries will be judged on ease of use, presentation, programming style and originality.

Till next time.

Neil.........

Included on the Coverdisk this edition are a number of files that go hand in hand with our tutorials. There are six examples of Assembly source code in the directory ASOURCE Also in this directory you will find a file called SOURCE.LHA, this is each of the examples as executable files.

In the directory marked LEARN_C you will find five example files that acompany the C tutorials (funny that). Finally in the

AMOS directory are two files called:
MINIPAINT.AMOS and MINIPAINT.ASC. One is an AMOS file the other is executable and they go with the AMOS tutorial (I like to repeat myself hehe).

DiskMate V3.2 is on the disk. It is a small disk copying utility that may be familiar to many. This is the latest new and improved version.

BootBlock Slide Show Maker has now had all the bugs fixed as is thus presented again.

BootPic by Nic Wilson will display a default or selected picture when the system is booted.

Degrader will let you retard your machine to operate software writen for previous versions of workbench.

IFF, also from Nic Wilson software this little file allows you to view pictures directly from workbench or the CLI/SHELL.

SuperFormat Version 1.0 has been written by Geoff Scheimann and it will format a disk with all of the regular directories already included.

SysInfo V3.17 is another great from Nic Wilson software. When run it interogates your computer giving back vital information such as operating speed, which versions of each chip are installed (Denise, Gary etc.) and available memory.

Virus Checker is the latest version from John Veldthuis in New Zealand. I use and recommend it.

Extreme Violence is a game for two players. There is no plot just get in there and kill each other.

MEDIA-FLEX

A new Australian Multimedia System makes it's Debut.

The World of Commodore Exhibition, held each July at Sydney's Darling Harbour Exhibition Centre, has earned a reputation for showcasing the very latest developments in computer hardware and software - particularly in the area of Multimedia.

This year's show will certainly be no exception, with an Australian produced Video Production and Multimedia System making its official launch. MEDIA-FLEX is a Non-Linear Editing system using the latest in digital video compression technology.

The manufacturers, Perth based Colour Computer Systems Pty. Ltd., claim MEDIA-FLEX to be the first PAL/NTSC Non-Linear Editing System to integrate Non-Linear ONLINE editing, 8 tracks of 16 bit CD quality digital audio, 2D and 3D graphics, titling, image compositing/morphing and DVE's into one system. It will provide video and multimedia professionals with broadcast quality digital video at a 4:2:2 specified resolution of 720 x 576 with outputs meeting CCIR601 requirements.

MEDIA-FLEX is also the first non-linear online editing system with full PAL broadcast resolution to offer networking to other systems and is equally at home as a multimedia resource facility feeding MAC, PC, AMIGA and SGI platforms. The concept of a digital video/graphics/audio network can be a reality with MEDIA-FLEX. Up to one and a half hours of SP BETACAM quality or several hours of VHS quality video can be stored with over 2 hours of CD quality audio. The base system will market for approximately

Don Sforcina

"MEDIA-FLEX WILL BE TO MULTIMEDIA, WHAT THE VIDEO TOASTER WAS TO VIDEO".

$40,000. A CDROM drive, scanner, DAT backup tape drive, removable optical disk, plug-in TBC and laser disk are some of the many options supported.

Two users may use the system at the same time, accessing the applications via two 20", 1280x1024 workstation screens. Each screen is further supported by a 24 bit framebuffer. Usage of a proprietary 32 bit Pixel Bus in combination with the digital video compression board enables realtime playback of 24 bit images at full video resolutions. MEDIA-FLEX incorporates three computers, two

powerful 68040 processors and one 160 MFLOP graphics processor allowing near realtime rendering of 3D graphics.

Don Sforcina, the company's Managing Director, a video producer with 20 years of digital systems research and development experience, describes these current digital video technologies as being "Just the tip of the iceberg, our MEDIA-FLEX system will revolutionize video and multimedia production". Other systems have limitations in the quality of their images, display resolution, storage capacity and video compression levels. MEDIA-FLEX is totally expandable and is designed to change with advances in technology, it can literally grow to meet the users requirements.

Colour Computer Systems P/L are marketing the JPEG and MPEG technologies for useage in OEM systems for Kiosk dosplays, in-store Point Of Purchase (POP) advertising, using modem transmission of digital video data, custom video disk applications and video telephones. A MPEG digital video/audio compression board will be released in the fourth quarter of 1993. In this product, the compressed audio data is an integral part along with the compressed digital video data. The extra production efficiency generated by MEDIA-FLEX will further enhance these applications.

Keep a good eye out for this system at the World of Commodore show, it will make you want one or you will turn green with envy.

The INFORMATION Age

By Neil McKnight

If the 1980s was the decade of the home computer then the 1990s is the decade of information sharing.

There are millions of computers in use around the world, and Amigas are still relatively rare in the workplace. You probably use an IBM compatible at work or school, but own an Amiga at home.

So, how can you continue working on that spreadsheet or assignment at home without buying a second computer or an emulator card?

Information Sharing

Now that Workbench 2.1 and 3 are available, every Amiga is now capable of reading, writing and formatting MS-DOS disks.

This gives all Amiga users the ability to transfer information between the IBM compatible you have at work or at school, and your Amiga at home, and without any special hardware.

For example, I use Workbench 2.1 on a standard A2000. My internal drive DF0: acts as both a normal Amiga drive and an MS-DOS drive at the same time. I just place a disk in the drive and Workbench works out what sort of disk it is, then displays the appropriate icon. No problem!

I can take files to and from work using the same disk in both computers. This lets me finish whatever I was working on, using an Amiga program, then take it back to work the next day.

Does WB2.1 Allow You To Play IBM Games?

Transferring information to and from another computer is not the same as using their programs. You cannot play an IBM game for example, unless you have a Bridgeboard or IBM emulator card installed. But you could copy and use an IBM disk full of documents, spreadsheets or database files.

Is Transferring Information Really Useful?

Yes! Many Amiga programs now directly import and export IBM data files. This is the real reason for information sharing.

The ability of reading the files created by another computer saves you from having to emulate that other computer.

Many people who go for an emulator card forget that they also have to set up their computer with another operating system, buy the programs they want to use, divide their hard drive space between the two systems etc. In fact, set up another whole computer just to run those other programs.

Information sharing is a form of compatibility that means you don't have to go to all that trouble. After all, most people who buy emulators already have access to the other type of system, otherwise they wouldn't need to emulate it.

What Can Be Transferred?

Text & Documents
Plain ASCII (text) files are the most common type of information that people want to transfer. Because there are no special formatting commands in the file, the text is easily read by any program that uses words.

Document files, on the other hand, contain extra information about the various modifications to the text such as bolding, underlining, different fonts etc. Such files may need to be converted to ASCII first.

Documents created on an IBM using MS-Word or Word Perfect can be directly loaded into many programs such as Prowrite, Wordworth, PageStream or ProPage without modification.

Data Bases
DBman, Superbase Pro and Amiga Vision are just some of the programs that can read and use the IBM DBase style data files.

DBase files are very popular on IBM compatibles. Most IBM programs that work with databases can export data in this format. As long as the two systems can find a format in common, you can transfer across to the other system.

Spreadsheets
Spreadsheet files from an IBM, such as those produced using Lotus 123, can be directly read by many Amiga spreadsheets.

Both Maxiplan Plus and Professional Calc could read and write Lotus files directly from an MS-DOS disk when used with Workbench 2.1, so you wouldn't need to transfer them onto a separate floppy.

Fonts

There are thousands of IBM Public Domain fonts that can be used on the Amiga. Some fonts for the Mac can also be used.

Programs such as PageStream, ProDraw and ArtExpression can use Adobe Type 1 fonts directly. Other programs supply conversion utilities to change them into a usable form.

Fonts from Windows and other systems can usually be converted to Adobe Type 1 fonts with a conversion program.

Graphics and Clip Art

Most desktop publishing programs now include special loaders allowing them to use the many IBM and Mac graphic formats directly. Several Amiga PD programs will convert them to Amiga IFF pictures for you.

Art Department Professional is one program that can convert almost any format into any other. Highly recommended if you use a lot of IBM graphics.

For true IBM compatibility, the ability to run an IBM program on an Amiga, there are a range of emulator cards to suit your needs and budget.

For occasional use, there are also some excellent shareware programs that emulate another computer through software. One such program is the excellent PC-Task written in Australia by Chris Hames. This program works best on an accelerated machine, but will function on all Amigas.
Check out your local PD library for others.

User Groups

Commodore Users Group ACT inc.

The aim of this group, is to promote the use of personal computers for educational and recreational purposes. We also provide a forum for the exchange of information and ideas amongst our members.

We support the 64, 128 and Amiga as well as any other Commodore machine.

We have two regular monthly meetings:

The first :-	**Melba High School** **Conley Drive** **Melba ACT** **on the first Thursday of the month**
The second :-	**Phillip College** **Launceston Street** **Phillip ACT** **on the third Monday of the month**

Meetings are informal and beginners are especially catered for. Demos and interesting talks are given.

We have an extensive Public Domain library and a large collection of computer books and magazines, so come to a meeting and have a look.

Send all enquiries to:

Commodore Users Group (ACT inc)
PO Box 489
Curtin ACT 2605
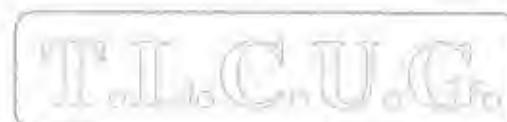
T.L.C.U.G.

The T.L.C.U.G. (Tuggerah Lakes Commodore User Group) was formed in 1985 by seven or eight C16, Plus4 and C64 users, during the ensuing years our membership grew to over 135 with a good cross section of varied knowledge.

The group owns a C64, 128D, 2 x A500s and a VCR, with PD and video libraries for each.

Our group meets on the first and third Thursday of each month. Meetings are held at KANWAL and begin at 7.30pm.

Send enquiries to: T.L.C.U.G.
PO Box 2257
Gosford NSW 2250

CHUG

Commodore Hornsby User Group

The Commodore Hornsby User Group supports Amiga and Commodore C=64 & C=128 users. Currently our membership is approximately 100, including some 80 Amiga owners. We meet twice a month at:

St.Leos College
Woolcott Avenue
Wahroonga NSW
at 7.30pm.

Workshops are held on the second Wednesday of the month and the main meeting is held on the fourth (not necessarily the last), Wednesday of the month, the main meeting usually features a professional demonstration of Amiga software and/or hardware.

We have various services available to members including:

- :- **PD Libraries**
- :- **Bi-Monthly magazine (Peripheral)**
- :- **Magazine Libraries**
- :- **Beginners Classes**
- :- **Phone Help**
- :- **Club Shop**

We have recently made changes to our BBS service and now have a doorway on the "Three Amigas" Bulletin Board (02) 609-4458. The Three Amigas supports bps rates from 300 to 9600, settings are F8N1.

Naturally visitors are welcome at our meeting as are callers on the BBS. The secretary can be contacted on (02) 487-1062 for more information.

REAL 3D

Version 2

carried out on the same screen. It is possible and sometimes necessary to use multiple screens, but the way in which they are used can be configured entirely by the user.

The highly configurable environment allows for the creation of user defined gadgets and single key macros to customise the entire user interface. The guys have expended a total of 4 man years developing this new version and it was definately worth the wait.

**Q.** What is the bottom line, how much is it going to cost me?

A. The program may seem expensive at first glance but that is only until you realise that REAL 3DV.2 can and does compete effectively with similar packages operating on Silicon Graphics platforms. Some of these packages like ALIAS, WAVEFRONT and SFT-IMAGE have retail prices that exceed $20,000. REAL 3DV.2 comming in at $899 is set to stun the entire computing community.

Q. What makes REAL 3DV.2 better than Imagine and it's equivalents?

A. To go into detail would take up more space than you have available, so I will try to be brief.

COWBOY (below)
TRACEY (left)

After my first glance at this program I knew at once that there was no possible way a review, written by anyone new to it's environment, could do it any justice.

It was at this time that I chanced to meet BRUCE BROWN, a long time beta tester for the authors of REAL 3D. Bruce consented to take part in an interview to answer a few questions.

I begun with the standard, everine wants to know questions and was quickly inundated by exactly how much I did not know! Anyway listed below are the questions I asked, along with the heavily compressed replies from Bruce.

Q. What is the minimum machine configuration I would need to run REAL 3DV.2 effectively?

A. The program needs a minimum of 3meg of ram, 5meg of hard disk space, a math co-processor and an 020 or better processor speed. This is due to the complexity of some of the mathmatical operations. I also recommend more than the minimum 3meg of ram.

Q. How does version 2 differ from earlier versions?

A. The first thing that most users of the earlier versions will notice is that the authors have dropped the idea of three separate parts to the program. In version 2 object modification, animation and rendering can all be

Textures: There are no fixed patterns, which gives you virtually unrestricted texture mapping properties. Images created with paint packages, video digitizers and even those scanned can be used to colour the surface of an object. REAL 3DV.2 treats textures as normal objects which means they can be modified or animated just as easily as any other object. It also means you can have an unlimited number of textures on any one project.

The RPL Language: REAL 3DV.2 includes it's own fully featured programming language which is very similar to FORTH. It is fully documented and is designed to be used for;

-: *Expanding the program features.*
-: *Macro recording*
-: *Scene description and saving*
-: *Particle and procedural animation methods*
-: *Procedural material properties*

Fractals: There is included a fractal tree and landscape generator. It was used to create the picture called sunset

SUNSET.

Hierarchy: Has full CUT and PASTE options and operating system operations like multi-gadget selection to assist in making the program even easier to use.

Broadcast Quality:REAL 3DV.2 supports full field rendering to give fast smooth animations.

Some of the other features include,

-Support for all picture formats
-Spline mapping
-New concepts like non-linear and three dimensional time
-Inverse Kinematics (usually only possible in programs like ALIAS and SOFT-IMAGE)
-Unlimited resolutions
-Boolean operations that can leave the texture of the knife on the cut surface
-Open architecture for ease of expandability
-Highly optimised ray tracing
-Quadratic surfaces

-B-splines
-A comprehensive set of free form surface construction tools
-Soft shadows
-Depth of field
-Motion blur
-9 Level anti-aliasing
-Revolutionary animation system based on object orientation

Manuals and Tutorials: The documentation weighs in at a staggering one and a half kilos. I have not been able to find anything the program can do that isn't covered in the manuals. Tutorials, well there is a tutorial for almost evey feature of the program, so learning how to use REAL 3DV.2 will be an adventure in itself.

There is so much more, I could go on and on for many hours. Suffice to say that any new package would need to be DAMN good to even come close to REAL 3DV.2.

Many thanks to Bruce for giving us his time...

Public Domain

In this edition we have been given a couple of games and a whole bunch of small utilities to look at. First off we have a text adventure which is of very good quality.

The Talisman is one of those PD games that really shows that good software doesn't have to mean big dollars.

Written in England by Paul Jenkinson Talisman rivals many of the commercial games with it's playability.

Your task is to find the lost Talisman, an object that was created to signify peace and union between the peoples living in two previously warring valleys.

One people created a fine gold chain, while the others brought forth a diamond of unusual size.

Together they made the Talisman. Lost when all those that guarded it were killed, this priceless object is the most sought after treasure in the realm.

Do battle with Orcs and Goblins as you search high and low, but always make sure you know where the closest water of life is located.

Begin your adventure in a cave, high in the north mountains and the best of Brittish luck to you.

XOper V1.3
This little utility is used to display and contol system activity and can be run as a background task. A very handy item for programmers from all genre.
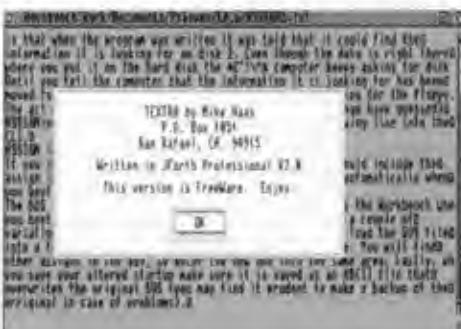
TEXTRA
A very easy to use text editor that makes no claim at being the fastest or the smallest or the any-other-est, it is just a very user friendly text editor.
This little program is just about perfect for all those small editing tasks.

CalcKey
This is a basic four function calculator intended to save you from having to hunt down your pocket job every time you need to do a simple calculation. CalcKey is a memory resident, pop-up program, which means it can be loaded into memory from the startup-sequence and called up with a simple key-stroke. The calculator will pop-up on whatever screen you are using at the time and is not restricted to flipping back and forth to the workbench screen.

This calculator is set up to do HEX operations as well, so could be used most effectively by programmers.

BezSurf
As you can see this utility creates bezier surfaces of revolution. It produces amazing pictures of wineglasses and doorknobs and any other object that could be turned on a lathe.
Here is the best bit, there is included in this program the ability to map IFF images onto the surface of any BezSurf created object.
This is a good program to play with as you can produce some truly startling effects.
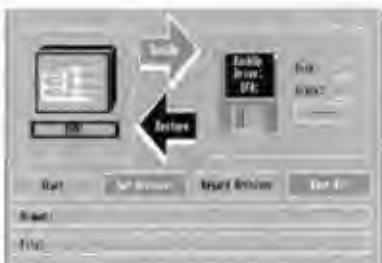
KwikBackUp

This is a hard drive back-up utility that makes use of the archive flag. It can easily be operated from workbench and is a must for anyone that has a hard disk. This program is fast and efficient.

Zap

Zap is a binary file view/edit utility that allows you to examine and modify a file in Hexidecimal or ASCII mode.

Zap keeps the file in memory, so moving about and searching is very fast. Because of this you will find that available memory will limit the size of file that Zap can handle.

Each edition I will have a look at various pieces of Public Domain software, if you have an intrest in a particular area of PD, please write in and let me know and I will endevour to find something wich suits your needs.

Write to:
Public Domain
PO Box 567
Mirrabooka
WA 6061

DSP

Digital Signal Processor

Commodore have decided to incorporate a DSP (Digital Signal Processor) in it's new range of Amiga computers. The chip designed by AT&T sits as sub processor but has a higher priority than the 680x0 processor.

The DSP, a risc chip running at 33 megaflops will most likely be faster than a 68060. The Amiga and Multimedia will be given a incredible boost when this chip starts being used.

The tech specs.

* 32 bit floating point arithmetic

* 32 bi addressing

* 8k on chip zero wait memory

* Single cycle instructions

* Shares bus with the processor

* Serial I/O with DMA transfer (serial data transfers occur without processor intervention)

* DMA control for serial in and out

* Barrel shifter for bit manipulation of graphics or data encryption
* Can control external hardware

* 32 bit timer for waveform generation

Some of the software coming out for it.

* Onboard v32 modem (software emulated to 9600)

* Fax modem (same as above)

* 3d graphic library

* Speech recognition (to some degree)

* Image and audio coders

* Midi music synthesizer

* Jpeg still image compression (Mpeg will follow)

* Sound processing

The Advantages of a DSP chip like this in multimedia are endless. Pictures compressed almost instantly, no need for modems or a fax. It also could be used to send images down the phone line aswell as voice.
All this and the 680x0 doesn't have to even lift a finger. The future of the Amiga looks very bright indeed.

ZAPHOD

VIRTUAL REALITY

Since launching Virtuality™1000SD - the worlds first virtual reality entertainment system, in March 1991, W Industries has made great strides in pushing the power of VR to it's limits with exciting new machines and challenging software experiences. Virtual Reality is a computer technology that immerses individuals in a 3D computer generated world.

where powerful software allows the player to work against either a computer opponent, or interactively with other human players.

Unique virtual reality centres exclusively using W Industries systems and software will be opening before the end of this year in Hong Kong, Korea, Switzerland, Austria,

Taiwan, Japan, Germany, Saudi Arabia, Israel, USA, the UK and best of all Australia.

A Perth based company known as Virtual Reality Entertainment Promotions has begun to introduce a number of the Virtuality systems into the country to extend the experience to Aussies who have been itching to have a go.

Keith Benson the General Manager of VREP shows us how to do the job properly.

There is a variety of software available (although limited at this point) to be experienced by anyone with an interest:

Dactyl Nightmare, a two player game set on five levels each connected by a staircase. Find your opponent and shoot him. Use available terrain to avoid getting shot yourself. Beware the giant Green Terradactyl that will carry you to a great height and drop you.

Capture the Flag, is the team version of Dactyl Nightmare. Adds a whole new aspect to sneaking around, shooting people. Guess what? More big Green Terradactyls than before...

Legend Quest, where you play the heroic knight in search of the lost talisman. Use your trusty shortsword to do battle with skeletons, wolves and goblins as you try to return the hard won treasure to your beautiful queen. Have a friend join you on your quest, he can watch your back and help fight your enemies.

XOREX, puts you inside a robot and releases you into a combat zone of the future.

Grid Warrior, is very similar to XOREX, except you are a Cyborg like the Terminator and you carry a minature cannon in your arms.

Flying Aces, lets you join the airforce in World War 1 and take flight in your trusty Bi-plane to have a little one on one with the Red Baron. This game also allows for other players to intrude on your air space, in which case you shoot them down (or try to).

Total Destruction, for anyone that has ever wanted to race in the NASCAR races at Daytona or Indianapolis and hasn't had the money or the opportunity, this is the closest thing to actually going there. Thunder around the track doing speeds in excess of 200mph, trying to hold it together long enough to reach the checkered flag.

VTOL, this time you will really have to keep your head. Climb into the cockpit, fasten your safety harness and get prepared to launch straight up into the wild blue yonder in the amazing Harrier Jump Jet. The Vertical Take Off and Landing aircraft is one of the most technologically advanced machines to inhabit our skys today. The chance to fly a virtual simulator will atract many a flight sim freak.

Battle Sphere, takes you from the Jump Jet into a space craft. Fly in and out of superstructure while you chase enemy craft around the Battle Cruiser you are trying to protect. Form part of a squadron and become heroes in a future far far away.

Star Trek - The Next Generation, start as an ensign on the star ship Enterprise and work your way up to the position of Captain Picard. Do battle with the vicious Romulan Empire. Sit upon the bridge of the most famous spaceship known to man(this is a new game and may not

Sue (28) Exellent, looking forward to next time!

Dale (11) Wicked, it was like being somewhere else.

WINNER

In last edition we had a competition to see who could find the most computer oriented words in our Word Find

The winner with the highest number of acceptable words was BRUCE McCREADY from Clifton Hill Vic with 68. Bruce wins $20.00

be available just yet, but it is not far away).

Virtual Reality Entertainment Promotions intend to open a Virtual Reality Centre in each state. Each centre is set to have four stand up machines (pictured) and four sit down machines. Some of the games mentioned are designed for sit down machines only eg; Total Destruction, Battle Sphere, Flying Aces and VTOL (XOREX is suited to both).

At the moment you can have the experience in Perth at 411 Murray Street for approximately 5-8 dollars. In other states there are a number of units travelling around to various places. In edition seven we will give more information about when and where you can have a go.

Each Virtuality unit has an approximate cost of $125,000 not including software. The

software bill runs to upwards of $7000 per game.

At the heart of the Virtuality system is a Commodore Amiga 3000 computer with the software stored on a CD rom drive. The A3000 is only used as a controller for the custom graphics boards within the system, these boards are what cause the unit price to be so high but they also make the graphics and animation rendering extremely smooth.

As mentioned earlier I will keep you up to date with the advances to our available VIRTUAL environment as they happen. In the next edition I will give the places and dates that these machines will be in each area.

I will also try to get pictures and more information regarding the new games and simulators you can have a go at.

Bye for now...

An Introduction To

Amiga

Hi there and welcome to the introductory BASIC column. I hope this column will become a regular part of the OZAmiga magazine. The aim of this column is to provide the basics of programming to the users out there who would like to learn but do not know where to begin. I will start with the basics and then go into more depth, depending upon your response, so write in and tell me what you would like.

About Basic

BASIC (Beginners All purpose Instruction Code) was developed in 1964 by John Kemeny and Thomas Kurtz at the Dartmouth College in

converted program is executed.
Why Basic?

The reason why I chose Basic is that nearly every Amiga owner already has a copy of Basic on their Extras disk (if you have a version of Workbench greater than 2.04 you won't, as I believe Commodore exchanged Basic for the Arexx language). This means that you don't have to fork out a couple of hundred of dollars on a interpreter or compiler as you would have to for another language.

The speed of Basic will not worry us as the programs you will be writing are reasonably small and thus will be executed at a respectable speed.

statements that can lead to badly written programs.
A First Course in Basic

As this is an introductory programming tutorial I will assume that you already know about Workbench, particularly how to open and load programs, if not have a read of the manuals that came with your Amiga.

AmigaBasic is located on your Workbench extras disk so first load Workbench then place your extras disk in your drive and load the Amiga-Basic interpreter.

Once AmigaBasic has loaded you

With David Perko

America. The intention of Kemeny and Kurtz was to create a computer language that is easy to learn, use and teach. Due to its simplistic nature Basic has become very widespread and is available on most home computers.

Basic is an interpreted language meaning that each time the program is run each instruction is converted to machine language and then executed. Due to this, Basic programs run fairly slowly as each time the program is run the conversion to machine code has to take place.

Most other computer languages are compiled rather than interpreted. Compiled languages first convert their particular code to a separate machine language program and then each time the program is run the

Basic has been criticised due to its nature allowing the use of badly structured code (i.e. code that is hard to understand and modify). In this column I will try to encourage people to write structured code that is easy to understand and I will not introduce

should see two windows, one labelled "Basic" the other "List". These two windows behave like ordinary Workbench windows, as windows are selected by clicking anywhere within the desired window.

As mentioned previously Basic is an interpreted language and the Basic window is the interpreter. Whatever you type in the Basic window is interpreted, then if the command is valid it is executed. Whereas the List window is where a set of instructions (called a program) is temporarily stored. From the List window the program can then be executed, stored on disk, edited etc. In todays column I will be concentrating on the basic window, next time I will introduce the list window and show you how to create, load and save programs.

The simplest basic command that produces anything useful by itself is the print statement. The print statement allows the user to display text on the screen. To use the print statement first select the basic window by clicking your mouse button anywhere within the basic window. To display the words 'Hello world' the following command is entered:

print "Hello world"

after pressing return the computer will respond by displaying 'Hello world'. The print statement displays on the screen whatever you type between the quotes(")

Play around with the print statement and see what happens when you place different text within the quotes.

If the above print statement is typed without the quotes (eg print Hello world) the compiler will assume that Hello and world are variables. Variables are a combination of characters that point to a place in memory where a certain value is stored. Since we have not defined "Hello" and "world" to be variables the computer will respond with two zeros when the command is entered.

To define a variable the following is entered into the basic window:

a = 1
b = 2

The above statements told the computer to set the variable 'a' to equal 1 and the variable 'b' to equal 2. If now we instruct the interpreter to display 'a' and 'b' by typing

print a b

the computer will respond by displaying a 1 and a 2.

To declare a string variable we have to include the dollar sign($) as the last character in the variable name, for example:

a$ = "this is a string variable"

In the above command we have declared a$ to be the string 'this is a string variable' so now if we type in:

print a$

the string that was declared to be a$ will be displayed.

As all string variables end with a dollar sign and all number integers

don't, an error will occur if we try to declare a number as a string, or a string variable as a number.

Note:
You can declare a number as a string variable if you include the number inside quotes, as anything inside quotes is taken to be a string eg:

a$ = "123"

The print statement will also allow mathematical computations, for example to calculate 1 plus 3 the following command is entered:

print 1 + 3

Once return has been pressed the computer will respond with the answer of 1 + 3 which is 4. If you would like the actual computation that was carried out to be displayed you should type in the following:

print "1 + 3 =" 1 + 3

If you typed the above command in correctly the computer will respond with:

1 + 3 = 4

Experiment around with other mathematical computations. The symbols for addition, subtraction, multiplication and division are:

+ = add
- = minus
* = multiply
/ = divide

In an attempt to make life easier for Basic programmers the designers have allowed programmers to use the question mark in place of the print statement eg.

print "hello world"

is equivalent to

? "hello world"

I have mentioned this shortcut mainly as a time saving device for people typing in programs with many print statements. My advice though, is not to use it as it is not a standard basic command and it decreases the readability of the program as a whole.

Now that you have learnt how to program the computer to output text to the screen I will show you how to program the computer to accept input from the keyboard. To demonstrate the input command type the following command:

input test$

After you have typed in the command and pressed return the interpreter will display a question mark and pause until the user has entered some text. The interpreter will not continue until the user has pressed return.

Once the above input command has been successfully executed what ever you typed in, the computer has stored in the variable test$. If we now print out the contents of test$ by typing the following:

print test$

the computer will display what you have just typed in.

Well that's it for today. To keep you busy till next time I suggest you experiment around with the print and input commands and have a read of the Basic manual. Remember the best way to learn a programming language is to play around with it yourself and learn from your mistakes. In the next edition I will explain how to write, save, load and edit your first basic program and show you how your Amiga can make decisions.

Until then,

David

ART and the COMPUTER

By Mark Johnson

HISTORY

The history of the computer in art is very much the same as the history of photography. It's been a slow, uphill process. The pioneering of computer graphics was in 1964. John Whitne Sr used a World War II M-5 gunsight which he adapted for experimental computer animation. The computer actually moved the gun sight around, which had a light source with a pin hole projecting light onto the film. It made nice patterns on a single frame.

In the 1960s programmers were printing out pictures using the characters of the keyboard. It was a long task that didn't produce great pictures.
It wasn't till the late 60's that the computer had images on the screen, or VDUs (Visual Display Units) Since then people have come to know the computer as an art tool.

TIME

I guess most people in the media world ask, "When will it be done?" To the traditional artist it could take a day to a year to complete a painting or an illustration. To the computer artist it could be the same. You say, "Why bother with the computer then if it takes the same time?" Well the computer as an art tool is still relatively new to this medium. If you consider what the micro chip has done to business and science, you can have a fair idea what the computer will do to the cinema and television field, and there is also desk top publishing that can take full advantage of 16+ million colours for pictures and

diagrams. Though 16+ million colours sounds a lot, there is still yet to be a colour printer that will have an output to match that won't cost an arm and a leg.

The computer artist can create a hi-resolution picture within 10 hours that looks impressive to the non-computer buff, and with time, photograhic looking pictures can be achieved with a paint box, or if you like a 2D format program.

If we look into 3D art, we are now talking time consuming. Digital Productions, one of the world renown computer graphic companies made around 8 minutes of computer generated animations for the film "The Last Starfighter". Using the world's most powerful computer (Cray XMP) it still took up to 6 hours to generate 1/24th of a second in some parts of the animations, and that was record breaking time!.

The time barrier for computer rendering is coming down all the time. People that waited 30 seconds for something now demand it to be done in 3 seconds. There seems to be something about humans, we want more things done in less time. I can see the computer as an art tool is certainly helping us and will do so more in the future.

COST

The cost of your local Cray XMP is millions of dollars, so it is understandable that you won't be purchasing one of those (besides, there's less than 30 of them world wide). If we consider that in the mid 80's TV stations spent half a million on a computer system to do their logos titles, weather etc and in the early 90's we have seen the Amiga come up to broadcast standard with 24 bit add on's at a fraction of the price, it won't be long before computers like the Cray or similar will be available for the serious artist. (See sidebar current 24 Bit Pal hardware products for the Amiga)

To be content at the moment is a painful task; we see high tech images in the cinema and say "wow, how do they do it?" and go away to try and reproduce it, to no avail. We are stuck with the manufacturers hardware and software guide lines. If we can't afford the high tech stuff, companies like Apple, IBM and Commodore with the Amiga range of computers have put boundaries on us. Though the Amiga is the ideal choice as far as cost and experience goes.

Of course there are thousands of computers on the market, and choosing one to suit your needs can be a daunting task. The range of software is dependent on the computer you buy.

COMPLICATING THINGS

With companies like P.I.X.A.R, who are most famous for their shot films, (Red's Dream, Nic Nac, Luxo Jr to name a few) create their own hardware and write the software to go with it. The advantage of this is, you're not stuck with the manufacturer's restrictions. Any problem they are faced with can be solved, just write another program. Most of the time they get the standard "flying logo" type of work, but on occasion someone will say, "I want a house transforming into a hotel while it is on fire showing smoke and flames devouring the building." Or more recently the movie "The Abyss" shown on TV a few months ago featured an

alien transforming water into a tentacle and then moving the water around the corridors of the rig. The disadvantage is you spend a lot of money writing new software for the customer. Of course in the process you are setting new horizions for the computer graphics field, which other people will soon copy from you.

The computer itself is complex, and when applying 3D graphics, it can indeed be confusing. Let's separate 2D and 3D.

TWO DIMENSION'S

2D paint programmes are basically an extension of painting. The exception is there are a wide and varied selection of tools at your disposal.

Tools to create circles, squares, curves, lines, there are brushes, stencils, mapping tools and the list goes on. (See sidebar for current PAL paint programmes for the Amiga)

often we will put in more than one. Now lets position the camera, (so we can see the ship) maybe add some stars and planets, or another ship in the background.

After all that be prepared to wait a couple of hours for the computer to render it. After the computer has finished, you take a look and... Aaarr no! That looks terrible, not what I wanted at all, back to the drawing board.
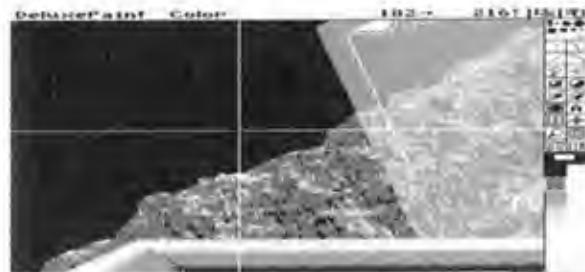
Many times my friends have seen my work and said, "How do you do it?" I often have a hard time trying to explain, which probably puts them off the idea of computer art. It's more a "Do it yourself" tool, than "Follow these instructions" by experimenting with the program you find out all sorts of tricks. If you apply yourself you can adapt to the computer quite easily.

THREE DIMENSIONS

The 3D world is more complex than 2D, and you can achieve better results in some instances. 3D art and motion is literal 3D. You have to create a 3D world in the computer. Look around you; you see chairs, tables, tv's, plants, roads, hills. As you see it, you have to create it.

We start with the modeling side. After building a model, let's say a space ship, you have to give the ship a colour, assign what it is made of and give it a reflection quality, if it has one. Maybe put some brush wraps on it to give it detail.

Ok, now we have to put in a light source, (where the sun will be) quite

After more changes the final result is done. (See sidebar for current 3D rendering software)

The A4000

Nic Wilson
Author of No-Virus

I have had my A4000 for a while now and have become quite familiar with it. These computers are available now, along with the new A1200's from all good retailers.

AA or AGA

Probably the most talked about addition to the new Amiga range is the AA or AGA chipset. The difference in the names here and the United Kingdom, is due to the name "AGA" already being used in the U.K. by Atari, so they changed it to AA. The letters stand for "Advanced Graphic Architecture".
This new chipset is the next logical step in the advancement in Amiga graphics.

Personally however I feel it may be a little too late. When I bought my first Amiga back in 1987, the graphics were far in advance of anything else on the market. IBM and compatible were just getting into EGA and it was very expensive and very slow. Apple and other firms were far behind and then did not have the co-processors that the Amiga had. Since that time five years have passed and the Amiga has stood still in this area except for a slight improvement with productivity, resolution and colours remained the same, and even to this date we have seen no improvement in sound capabilities at all. The sound in the A4000 is identical to that in the A1000 with the exception of the switchable sound filter that was added in the A500. During this time all other computers have advanced their graphics and 256 colours is common place in all games on other computers except the Amiga. I'm sorry to say that most other computers are better than the Amiga in their graphics and sound capabilities and there are thousands of software titles supporting their modes, whereas the Amiga has brand new graphics, old sound and no titles available at all yet. I get really annoyed at the fact that my 386 PC has better games than my 4000 or my 600, and there seems to be less compatibility problems with the PC.

In Amiga terms though, the "AGA" chipset is an advancement. Contrary to popular belief, CHIP memory remains and is not any faster. The new custom chip, "ALICE", however has faster access to it, four times the bandwidth in fact in the A4000 and twice the bandwidth in the A1200. (ALICE is the new version of the older style AGNUS chip). The limit to CHIP memory is still two megabytes but obvious signs on jumpers on the 4000 motherboard shows this has been designed to be increased to 8 megs if and whenever a newer version of ALICE is released. The limit of 2 megs is set within ALICE. Also contrary to some rumours I have heard, ALICE cannot access FAST memory, not even in the new modes.

What about the colours I hear you ask. This has been improved by increasing the number of colours available on screen at once and extending this so that every mode is available in all resolutions. This means that modes such as HAM, or the newer HAM8, are available not only in LOW RES, but in HIGH RES, HIGH RES INTERLACE, SUPER HIGH RES, SUPER HIGH RES INTERLACE etc. This goes for all modes, they are all available in all resolutions. This is a large improvement over the older chipset. Link this fact with the increase in access to chip memory and you get the older modes much faster, new modes in usable speeds and many modes not available before, not even on a PC with SVGA. In theory with SUPER HIGH RES INTERLACE in HAM8, you could have around 640,000 colours on screen at once, from a palette of 16.7 million, but it is better to consider the new chip set as having 256,000 colours at once from a palette of 16.7 million. When I compared HAM8 to real 24 bit output I really could not tell the difference, HAM8 is extremely good and does not suffer from the fringing problems of the older HAM6 mode. With some programs that are already available on fish disks such as "Viewtek" you can even view 24 bit pictures directly without having to convert them first. Other programs already working with the new modes are programs such as Art Department Professional, Imagemaster, Viewtek, my own "IFF" program, Rend24, UChess to name a few, this list is growing as software houses jump on the AGA bandwagon. I hope that this explodes at a tremendous pace when the A1200 is in plentiful supply. The new modes really make a lot of the 24 bit cards rather obsolete for displaying 24 bit pictures when HAM8 on a standard A1200 or A4000 does just as good a job.

The interlace modes, have a down side compared to the A3000. The de-interlacer is a software based design that is handled by new custom chips and a multiscan is required as some modes are not SVGA compatible as they were on the A3000.
This means that where a $300+ monitor did a good job on the A3000 you cannot use such a monitor successfully on the A4000 or A1200 as some modes will switch down to frequencies that SVGA monitors cannot handle. One amazing fact too, is that most new multiscan monitors no longer handle these frequencies either, so one must pay $800 - $900 for a Commodore monitor or try to get an older multiscan such as an NEC 3D for it to work successfully on the A1200 or A4000. AmigaDOS V3.0 does have a "VGA ONLY" monitor icon to allow SVGA monitors to be used but this severely affects your choice of resolution modes and virtually knocks out half the improvements you have paid for. I believe that this fact alone has the capability of severely affecting sales of the A1200 as it pushes the price over the $2000 mark, and this amount of money today buys a very impressive PC system, that Amiga salesmen will have to contend with. The A4000 is another story of course, but with a price tag of close to, or over $5000 with a monitor it is going to have to compare with fully blown 486 systems, with CD-ROM drives, massive cache memories, super fast CPU/FPU speeds etc. (that sell for around $2000).

If we forget all this and look at the A4000 for what it is, it is a very impressive Amiga and performs better than expected, although many developers are having severe overheating problems with theirs. I believe that this has been linked to cooling problems with SUPER GARY, that is situated underneath the A3640 card. I also believe that the engineers are currently working on a fix for this problem. My own A4000 has performed flawlessly,

although I did perform some modifications to the cooling design on mine as soon as it arrived from experience I gained with cooling problems in the A3000/PP&S 68040 card. Whether or not this is the reason I have not had any problems with my own I do not know.

One of the first noticeable differences in the A4000 is its appearance, the front panel and keyboard are white, very different from older models. A keylock has also been fitted to the front panel that locks out the functions of the mouse and keyboard. The mouse must be carefully placed when this keylock is enabled because the mouse buttons remain active. The keyboard is not interchangeable with older models due to the change in the mini din plug fitted to the end of the cable, this is however, exactly the same as the CDTV and the A4000 keyboard can be successfully used on the CDTV. The mouse is very smart looking, but probably the worst one Commodore has ever shipped. Failure of the mouse occurred within two weeks of use and the local dealer is having a 100% fail rate of A600 mice, after short periods of use, and these are exactly the same as the A4000 mouse. Personally I retained the "pregnant" style mouse from the A3000 and I am still using it on the A4000. It may be wise for users to consider purchasing another mouse instead of using the one supplied.

The A4000 comes fitted with 6 megs of memory configured as 4 megs of FAST and 2 megs of CHIP, and all memory including CHIP memory are fitted as SIMM type memory. The CHIP SIMM is of course a 2 Meg SIMM configured as 512K by 32 bits. The FAST memory SIMMs are 4 Meg types configured as 1 Meg by 32 bits. They are compatible with standard IBM PC SIMMs of 1 Meg by 36 bits as the Amiga ignores the extra bits. The latter will probably be the easiest to find and the cheapest. The hard disk is a Seagate ST3144A 125 meg ultra slim IDE drive. Yes IDE, SCSI has been dropped and you will have to purchase a separate SCSI card for one of the Zorro slots if you require SCSI. This was of course a cost cutting decision and I believe it to be the most foolish decision Commodore has ever made. They should have given the user the option of either SCSI, IDE, or both and set different price structures accordingly. This could have been done by leaving the controller off the mother board entirely and fitted another slot somewhere to fit a controller.

The A4000 comes fitted with a single 3.5 inch High Density floppy drive that can be formatted to 880K or 1.76 meg. These drives must not be confused with IBM 1.44 meg high density drives as they are not the same and not interchangeable. Memory has the ability to be expanded to a total of 16 megs of memory by standard JEDEC 4 meg SIMM modules as used by IBM PC's. Special ones can be manufactured that can take advantage of the fully fixed and working static column mode that the A3000 had, but in a non working form.

The design of the case is much better than earlier designs and is much easier to gain access to. The front panel always remains on the unit and the lid can be taken off by removing two small screws from the back. The power switch is on the front panel but has been indented so it cannot be accidentally

bumped off as it could on the A3000. There is room inside the case for another hard disk, the same size as the supplied one. There is room in the front panel for another floppy drive as long as it is one of the third height types. There is a 5.25 inch bay in the front panel that will JUST fit a standard size floppy, hard disk or tape drive with a push. There are plenty of spare power connectors inside the A4000, much different from the A3000 that lacked in this area. The orientation of the drive bays is rather strange. The floppy drive that comes fitted is below the 5.25 inch bay and also below the height of the keyboard making it difficult to insert disks if you have the keyboard directly in front of the computer. I personally believe it would have been better to have placed the floppy bays at the top.

Zorro slots are configured the same as the A3000 except the video slot has been moved to the bottom slot instead of the top one, and Amiga audio been made available to this connector to obviously allow external and internal audio to be mixed. The A4000 daughter board and mother board has been re-designed so that the unit can be operated without the daughter board being fitted, this is much better as I found this a real nuisance when repairing A3000's, although most users would never know the difference.

The A4000 is fitted with the FULL Motorola 68040, and not the EC version as specified in the manual and as published many times. ALL A4000's to date, have shipped with the full version including the Memory management unit (MMU) and floating point unit (FPU). The processor is clocked at 25 MHZ, but do not be fooled by this apparent slow clock speed, the 68040 running at 25 MHZ will blow away any 68030 running at 50 MHZ no matter what your mate that has one may say, and the internal floating point unit leaves the 68882 standing at any speed on 68040 optimised code and is still at least twice its speed emulating the 68882.

Changing kickstarts on the A4000 has been made easier by the inclusion of special hardware on the CPU card designed for that purpose. Commodore however decided not to release the software to enable it. I extended the ability of my Set040 program to include this function. This means that you may launch any kickstart you desire on the A4000 and still retain full compatibility with the new Commodore 68040.library, as well as being able to FASTROM the shipped kickstart. I have released Set040 to shareware and the suggested donation is $35.00 and is available from myself. (See details at the end of this document.)

The design of the motherboard is interesting and obviously saved Commodore some cost. The motherboard was designed without a CPU at all, and only had a FAST SLOT fitted which is compatible with the A3000T. Into this slot the A4000 ships with an A3640 card. This is exactly the same card that can be purchased for the A3000T. This is a good idea, as it means the A4000 could be accelerated by "exchanging" the CPU card to a faster one, etc., without the older CPU remaining behind useless as is the case in the A3000 design. This also paves the way for other models based on the A4000 mother board and makes them easily upgraded.

Below is a listing of the resolution modes available in the A4000 and the corresponding vertical and horizontal frequencies, and minimum size with no overscan.

Display mode

Size	Horiz	Vert
DBLNTSC High Res		
640 x 200	29.02 kHz	59Hz
DBLNTSC High Res Laced		
640 x 800	29.02 kHz	59Hz
DBLNTSC High Res No Flicker		
640 x 400	29.02 kHz	59Hz
DBLNTSC Low Res		
320 x 200	29.02 kHz	59Hz
DBLNTSC Low Res Laced		
320 x 800	29.02 kHz	59Hz
DBLNTSC Low Res No Flicker		
320 x 400	29.02 kHz	59Hz
DBLPAL High Res		
640 x 256	29.45 kHz	50Hz
DBLPAL High Res Laced		
640 x 1024	29.45 kHz	50Hz
DBLPAL High Res No Flicker		
640 x 512	29.45 kHz	50Hz
DBLPAL Low Res		
320 x 256	29.45 kHz	50Hz
DBLPAL Low Res Laced		
320 x 1024	29.45 kHz	50Hz
DBLPAL Low Res No Flicker		
320 x 512	29.45 kHz	50Hz
EURO72		
640 x 200	31.43 kHz	70Hz
EURO36Hz High Res		
640 x 200	15.76 kHz	73Hz
EURO36Hz High Res Laced		
640 x 400	15.76 kHz	73Hz
EURO36Hz Low Res		
320 x 200	15.76 kHz	73Hz
EURO36Hz Low Res Laced		
320 x 400	15.76 kHz	73Hz
EURO36Hz Super-High Res		
1280 x 200	15.76 kHz	73Hz
EURO36Hz Super-High Res Laced		
1280 x 400	15.76 kHz	73Hz
EURO72 Productivity		
640 x 400	31.43 kHz	70Hz
EURO72 Productivity Laced		
640 x 800	31.43 kHz	70Hz
MULTISCAN		
640 x 240	31.44 kHz	60Hz
MULTISCAN Productivity		
640 x 480	31.44 kHz	60Hz
MULTISCAN Productivity Laced		
640 x 960	31.44 kHz	60Hz
NTSC High Res		
640 x 200	15.72 kHz	60Hz
NTSC High Res Laced		
640 x 400	15.72 kHz	60Hz
NTSC Low Res		
320 x 200	15.72 kHz	60Hz
NTSC Low Res Laced		
320 x 400	15.72 kHz	60Hz
NTSC Super-High Res		
1280 x 200	15.72 kHz	60Hz
NTSC Super-High Res Laced		
1280 x 400	15.72 kHz	60Hz
PAL High Res		
640 x 200	15.60 kHz	50Hz
PAL High Res Laced		
640 x 400	15.60 kHz	50Hz
PAL Low Res		
320 x 200	15.60 kHz	50Hz
PAL Low Res Laced		
320 x 400	15.60 kHz	50Hz
PAL Super-High Res		
1280 x 200	15.60 kHz	50Hz
PAL Super-High Res Laced		
1280 x 400	15.60 kHz	50Hz
SUPER72		
400 x 150	24.24 kHz	72Hz
SUPER72		
800 x 150	24.24 kHz	72Hz
SUPER72 High Res		
400 x 300	24.24 kHz	72Hz
SUPER72 High Res Laced		
400 x 600	24.24 kHz	72Hz
SUPER72 Super-High Res		
800 x 300	24.24 kHz	72Hz
SUPER72 Super-High Res Laced		
800 x 600	24.24 kHz	72Hz

All the above modes support 256 colours when used on the Workbench Screen. Colours can be chosen in groups of 2,4,8,16,32,64,128, or 256 colours.

The A4000 like the A1200 comes supplied with AmigaDOS V3.0. This new version is a better version than V2.04 or 2.05 were. The main differences are full support for the AGA chipset, the addition of LOCALE and CROSSDOS and the improvement of many workbench and DOS programs. I had forgotten just how primitive V2.04 still is when you have been using V3.0 for such a long time. The books supplied are now split into several manuals and are A5 in size, much easier to handle and read than the cumbersome ones supplied with V2.05.

Locale is not of much use to English speaking countries but does make the Amiga much more professional and brings it more into line with systems like the Macintosh or IBM PC. Briefly Locale, localises your set up for the particular area of the world you live in, giving the user the language, special characters and string formats etc, he or she is used to.

The Amiga now has full support for postscript printers, allowing you to output directly to postscript without having to go through a program such as PageStream. Other programs such as Palette have changed to include the ability to customise the use of colour within windows and menu text and backgrounds. Fountain has gone and has been replaced with a similar program called Intellifont. Format has been extended with the ability to recognise and format high density disks. The addition of CROSSDOS means the user can directly interchange files between MS-DOS and AMIGA-DOS and format the former as well. There are many more changes in AmigaDOS V3.0, but the ones listed would be the most outstanding.

For those users who are into animation and use DCTV, you will be disappointed because PAL DCTV's do not work on the A4000. John Rowe and myself have been in contact with Digital Creations and they are currently working on the problem. It does not seem to be an AGA fault either because they work fine on an A1200. I would be very interested in anyone who has any success with a DCTV on an A4000.

CONCLUSION

So looking at all this, the new A4000 is faster than most others except for a few that are using faster 68040's with 40ns memory, but even then the A4000 is still much faster than these when it comes to graphics speed. It sports enhanced graphic capabilities, a brand new kickstart/Workbench and a high density drive, all considered it compares very well with an IBM PC compatible but it is lacking in software to take advantage of the new modes. It is much more expensive than a similar IBM PC compatible system and it requires a special and expensive multiscan monitor. The Amiga enthusiast who can afford it will obviously rush out and buy one, but we need to pull users away from other brands to increase the popularity of the Amiga. To do this the price must comes down, a de-interlacer card to make all modes SVGA compatible and a load of AGA software must arrive on the scene real soon.

Regards to all my readers and users

Nic Wilson
P.O Box 1164
Toowoomba
Queensland 4350
Australia
Ph: (076) 358539

GUILD HALL

Welcome back heroes and heroines to another edition of monster bashing and problem solving. This month I've got news of an adventurer in trouble, one Robin Criddle, who is having trouble with a few things in Ultima V. So we will start by answering his questions as best I can.

Robin wants to know why does he have to get the Sandalwood Box and where can he get his hands on Skull keys? Well Robin old son, it has been nearly three years since I solved the riddles in Ultima V but after consulting a few wizened old adventurers and looking in my trusty note books, I can tell you with only minor fear of contradiction that the Sandalwood box has to be in your possession or Lord British will not appear and guide you into the

Now on the adventure front this month is a bit of a trip down memory lane for those of you who can remember playing games on machines like the Vic 20 and the Commodore 64. Infocom have released a package of their most revered adventure games - adventure games that are text only and are in some ways better than the full graphic games we play so much today. There is something about having to imagine every detail of a scene and then trying to remember where you are, that is lost when the details are drawn for you - it is never the same detail and lacks individuality.

I'm not knocking graphic adventures but once in a while it is nice to slip back into these old adventures and give the old imagination a good work out.

a computer owner for hours on end.

Now while not a new release, Legend Of Kryandia gets a mention this month purely because it hasn't been mentioned before. This game is pretty basic, I would describe it as an adventure game with built in training wheels. It starts out easy - so you can get warmed up - and gets progressively harder as you go on. The puzzles are enough to keep you interested but nothing very hard if you just look at it thoroughly.

The key to this game is your mouse. Nearly everything you want to do can be done from the mouse and clues to what you should do can be gained from watching what the mouse lets you do. I'm not a great fan of this game, although it is interesting to note it is done by WestWood Associates - the people who bought out the great Eye Of The Beholder games. Legend Of Kryandia is definitely different.

Stay tuned next time for details on a great competition where you can win the latest adventure game. I warn you this competition is not for the faint hearted and will require a bit of brain power to
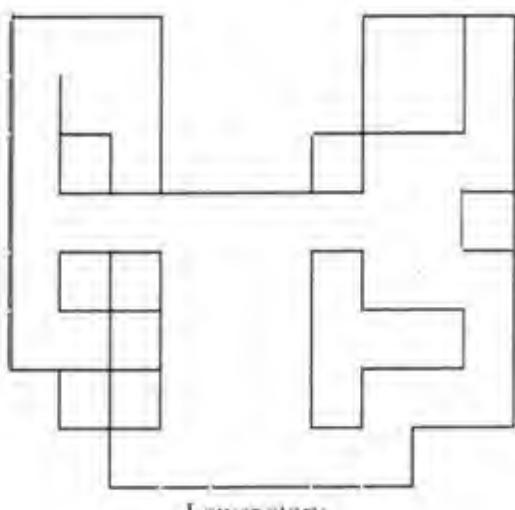
Lower story
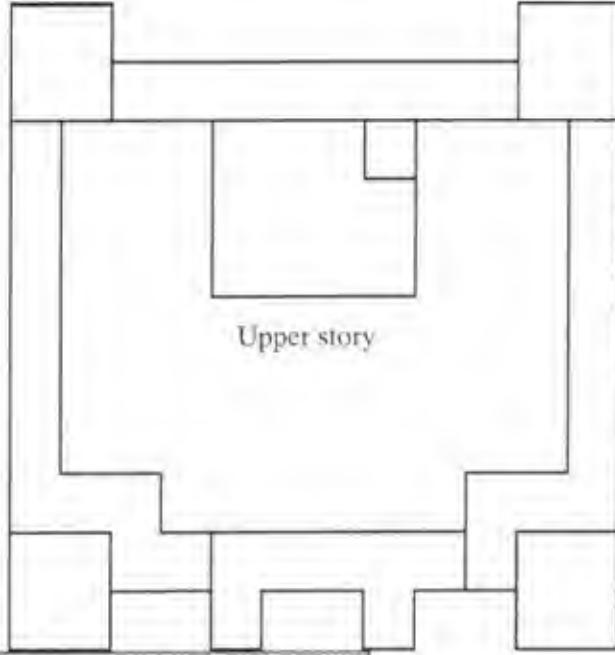
Upper story

From Death Knights of Krynn comes this map of **GARGATH OUTPOST**

underworld. Apart from that I couldn't find a use for it from memory - has anyone else got a use for it?

The skull keys? Well you can find a few if you look in tree stumps and I am pretty sure the secret guild of thieves sell them - but you have to find their stores and you have to meet their price. I think there are some located in the lower levels of Lord British's castle but I'm not too sure.

The package will also appeal to that special breed of adventurers out there who live in a text adventure world - Zorkers. The classic Zork adventures are included in this package and if you have never played Zork then you have never lived. Forget Eye Of The Beholder and Dungeon Master, these are the original dungeon exploration adventures that have captivated many

get anywhere.

So until then keep sending in your problems and I especially want you to send in anything of interest you find out while playing any adventure or role playing games, you never know someone out here might be stuck just where you are.

Guild Master
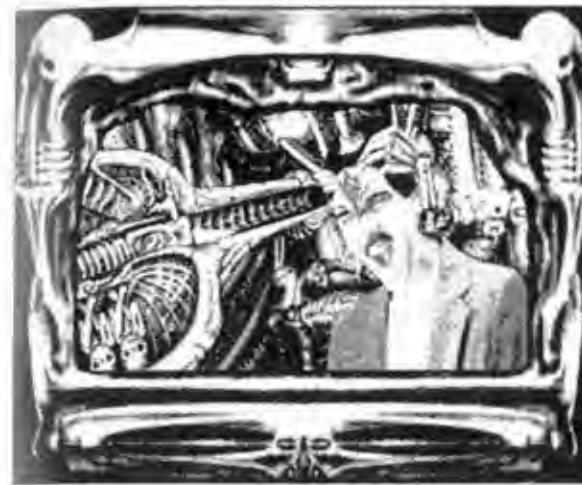
DARKSEED

Dark Seed is an interactive role playing game with a touch of the macabre. Incorporating artwork by H.R. Giger, who's oscar winning work on Alien, Alien II and Poltergist II changed the look of science fiction forever. His unique style depicts the synthesis of technology and biology as they might evolve without the influence of man. This game will have you on the edge of your seat.

You play the role of Mike Dawson, a science fiction writer who just

Look at everything with your character because if he doesn't see it then the game assumes that you haven't seen it either and many of the clues that lead you on in the game are hidden within objects.

Control the fate of two worlds... The world as we know it and the dark world of an ancient civilization. Unlock the secret of a sinister plot and dicover the dark passage to their world.... A place more terrifying than your darkest nightmare.

Time is running out.... You are on a collision course with destiny and only you have the ability to save yourself and the world from a cruel and inhuman fate. Every decision may be your last, so choose wisely....

Gruesome real time animations abound depicting terrifying nightmare sequences which help to keep you on edge.

Everything is controled by the mouse in this adventure that spans six whole disks. Dark Seed will run under both versions of workbench and is hard disk installable (thank god). It contains some very knotty puzzles and will keep everyone pulling out their hair until we have a bunch of bald Amigans around the country.

I recommend this game to those with time on their hands who like to indulge in a bit of lateral thinking.

Reviewed by: David Reeves
Supplied by : Headlam Computers
RRP : $69.95

My head feels like it's going to explode?
Boy my head hurts?
My head is killing me?
My head feels like it's goi
explode?
Boy my head hurts?

purchased an old victorian style home. The game begins as you wake up in your bed in your new house. Through the night you have been plauged by horrible nightmares which have left you with a splitting headache. You must of need, find the bathroom and get some medical relief, QUICKLY.

As you explore you soon discover that you have bought more than you bargained for.

Search the rest of the house and surounding areas to find useful objects and interesting bits of information. In this game everything is not as it seems, so take nothing for granted.

Your nightmares get worse!

Curse of Enchantia

Adventure and role playing aficionados will cringe as yet another adventure parody hits the shelves with enough dry humor and wit to tickle the funny bone of everyone who has enjoyed the odd Monty Python sketch.

said that, you sure don't have to wait long for the different parts to load so the lack of hard drive support is not a problem.

possible for hidden coins. You can't see them even if they are there but pick them up when you find them.

Our intrepid hero starts the game hanging upside down on a dungeon wall with no visible means of freedom - this is the first puzzle.

What else do you do in a situation like this? You yell for help of course! Our hero makes so much noise that the warder comes in and tells him to "QUIET!!!" and proceeds to drop the key on his way out. Now that I've told you how to start the rest is up to you.

The next room has some nasty little surprises - try looking in the keyhole of the door. And watch out for the jewels and things lying around.

The giant grey jelly bean with the mean looking axe is not a friend of yours either so be careful of him. And if you want to survive the bit after this then you better remember to take the goldfish and bowl with you.

I would say that Curse of Enchantia is a light - hearted and funny adventure game that appeals to almost everyone and will keep you going till you have it finished. It is one of the better releases of 1993 and heres to hoping the standard only gets better.

Reviewed by :
Bill Holder

Supplied by :
Headlam Computers

RRP : $69.95

Just hangin around

The game is set in yet another mystical medieval country with its quota of dungeons and monsters and traps and weird countryside. The whole game has a comic. light - hearted atmosphere. The graphics - while not being the best - are well suited to the game and the sound effects are very good, sometimes even quite funny.

Now one thing I don't like about this game - or any other game that does the same thing - is that it only uses the internal floppy drive and cannot be installed on the hard drive. For a six disk game not installing on the hard disk is bad but not using available external floppy drives is unforgivable. Having

The game has some very tricky puzzles and is designed in such a way that it is near impossible to continue on until you have solved the puzzle you are up to. A hint here - check in the bottom right corner as often as

Dangerous situations

BLOOD MONEY

Pause the game and press 'HELP' for infinite lives.

BLUES BROTHERS

On the player screen, type 'HOULO'. You are now able to select any level by pressing keys 1-6. Press space to begin play at your chosen level with infinite lives.

CARRIER COMMAND

Pause game and type 'THE BEST IS YET TO BE' followed by the '+' key. This gives you indestructable mantas.

DARKMAN

There is a cheat activated by typing 'MEACULTA' whilst playing but I have no info on exactly what it does.

DAYS OF THUNDER

Pause the game and type in exactly this 'COMEFLYWITHME'. It will really make you fly.

DRAGONS LAIR II

To make things a bit easier type in 'GET MORDROC DIRK' before you start.

EYE OF HORUS

Instead of pressing fire to start game, type in 'SPAM'. This gives you infinite lives and removes the need for coloured key to use elevators.

GODS

Type 'SORCERY' in the password section to get unlimited energy. This should also work on the main screen whilst playing.

HARD DRIVIN 1&2

Get your car up to a high speed and type 'N', this puts the car into neutral. You can now coast along at top speed with easy steering, invulnerablity and no time limit. Should work with both versions of the game.

HUDSON HAWK

Type 'SCIENCEFICTION' into the title screen to get unlimited everything. The 'DEL' key allows you to skip levels.

NITRO

To get 5000 fuel points and 50 credits, enter your name as 'MAJ' at the beginning of the game.

POPULOUS II

```
060 - INUNAD
090 - UXII
190 - JIERAF
220 - DDLLAB
280 - VEEGAD
320 - DONEAG
350 - GHOMAD
399 - LEQUAD
446 - JIIIAT
530 - TUUXAT
560 - PEITAG
610 - ABTTAG
650 - UMSIAC
690 - ERDDAF
730 - UXDOAB
780 - UBGHAD
840 - OPITAK
900 - OMUBAG
690 - DOAC
999 - WOITAB
```

IMORTAL

```
Level 2 - BEFE810006F70
      3 - CC5EE21000E10
      4 - 465FA31001EB0
      5 - B57F943000EB0
      6 - 1BBEB53010A41
      7 - 8DDFB62010AC1
      8 - E011F730178C1
```

ROBOCOP II

If you are not getting anywhere type 'SERIALINTERFASE' into the title screen to access cheat mode. Now pressing F9 will restore energy and F10 will skip levels.

RUNNING MAN

Get into the high score table and enter your name as 'DdIiSsKk' this should give you infinite energy.

SWITCHBLADE II

On the title screen, type in 'LEVEL' followed by the level at which you wish to begin play. You can also type in 'CHROME' to enter a hidden sub game.

CHROME

1 - no code	2 - truth
3 - jelly	4 - story
5 - cloud	6 - mouse
7 - human	8 - ?
9 - paper	10 - earth

LOTUS III
THE ULTIMATE CHALLENGE

Lotus III comines all of the best elements of the first two lotus challenges, as well as offering a number of new features to put control firmly in the hands of the player.

The expanded options allow you to alter the majority of features to suit your personal tastes.

There are five new themed levels, each with it's own distinctive graphics style added to the new handling features to make this an excellent follow up to LTC III.

Drive the M200, a new lotus concept car that will

never actually go into production. This means that the only way you will ever get to drive one is in this game.

On top of all that, there is a host of new sound effects. The most startling addition however is the 'RECS' system which gives you access to a practically infinite number of different tracks (the actual number is somewere in the vicinity of ten followed by

twenty three zeros, but whos counting).

Unlike a standard track editor which

may take anything up to an hour to design a single

track, the 'RECS' system will let you design a track in as little as a minute. RECS also saves the track code so that you can instantly pass your track design onto your friends.

As car simulators go, the Lotus series has been very well recieved and this latest edition should be no exception.

FLASHBACK

© 1992 DELPHINE SOFTWARE INTERNATIONAL/US GOLD LTD

It is finally here!

The long awaited sequel to Another World. The game that fired the imagination of many a game player. Using stunning graphics coupled with realistic actions and movements, Flash Back is destined to be another such hit.

After the intro animation you find yourself waking up from a stupor with very little memory. As you wake you

notice that you knock something with your elbow. Going down the deep crevasse you retrieve a holocube, which when activated shows you a picture of yourself. You tell youself to meet someone in the city, "yeah what city?" you say looking around at the desolate jungle.

Using all of your considerable skills you must delve deeper into the holes and tunnels around you to find the hidden (and protected) way into the city.

Located around you, at stratigic places, are elevators. These are the key to which way you must go. Sometimes they lead to an important object like a shield recharger or maybe just to the next room.

Your character is able to perform a variety of moves including numerous types of jumps, a body roll landing with weapon ready and sneaking into the next room with weapon at the ready. You will need to master each move with much practice so you can defeat your enemies.

When you reach that far away city, the truth is revealed and you need to find a way to get to Earth. There is only one way to do this, WIN at the 'Tower of Death', a game show where you are the entertainment. The fun has only just begun!!

This is the ultimate game players game, it works under both versions of workbench, requires 1 meg of ram and is much smoother on accelerated machines.

Thouroughly recommended.

Reviewed by : David Reeves

Supplied by: H e a d l a m Computers

Dear Denise

Dear Denise,

I have read all the reviews on the new AMIGA 1200 and it certainly sounds like a great machine to upgrade to. However, can anybody at OZAMIGA magazine tell me if the 1200 is compatible with my two favourite games, Strip Poker I and Strip Poker II.

Heinz (Beans) Vainker. (NZ)

Dear Heinz, or may I call you Beans.

I hope that it is not you and all your half-cousins down there in the Land of the Long, White Whatnot that are causing the disastrous hole in the Ozone layer. Please cross your legs 'til further notice. My sources inform me that the new AMIGA 1200 will run Strip Poker I and II, but apparently the girls take their clothes off so quickly that you don't get a chance to see anything !

Sorry Pal.
Denise.

Dear Denise,

I have recently come across a couple of references to a "Credit-Card" slot with regard to the AMIGA 600, and lately the new AMIGA 1200. What exactly is the function of this slot?

Albert Facey. (WA)

Dear Albert,

This is a rapidly developing area of computer technology. The "Credit Card" slot or more correctly the "PCMCIA" slot, an acronym for Personal Computer Monetary Convention with Instant Access, is only partly implemented in the AMIGA 600 and 1200, and can give no more than an Account Balance on supported cards (currently ANZ, Westpac and MasterCard).
However, I am assured that the soon to be released AMIGA 2200 will be capable of emulating a fully functional Bank Auto-Teller Mach- ine. Well done COMMODORE !

Yours in arrears.
Denise.

Dear Denise,

I am currantly in the market for a good Desktop Publishing package to produce a weakly Newsletter for the Train Spotting Club of which I am the secretery. However, I am rather septic regarding the inflated claims which the various contenders make in there advertisements; I am convinced that they are all invertebrate liers.
Please recommend the best value software for my system which is an AMIGA 1000 (512K), OKI 9-pin printer and Rank Arena 18 inch colour tellie.

Yours Faithfully.
Frank Althwaite. (SA)

Dear Frank,

Thanks for your letter, in which you seem to be having a little trouble with your worms.
I think that your first priority should be a Word Processor with a Spell-checker and a Thesaurus. And I am not suggesting that you go out and dig up a large prehistoric reptile.

Yours in Dislexia.
Denise.

Dear Denise,

My flatmates and I have read the reviews of the new AMIGA 1200 with great excitement; however we are still holding back from upgrading from our current Sinclair Spectrum as Nigel says that the ATARI Falcon will blow the 1200 right out of the water (excuse language). We have written to you in the hope that you will give us an impartial opinion.

P.S. We are all gay.
Simon Bumley. (ACT)

Dear Simon,

You certainly sound very happy from your letter. My advice to you is to just remember the old saying, "Smile and the world smiles with you, buy ATARI and you frown alone".

Yours cheerfully,
Denise.

Dear Denise,

Just a short note to say how much I appreciate your column in OZAmiga. I'm sure there are lots of novice AMIGA users like myself throughout Australia who are helped enormously by your heart-felt advice.

Dick Hedley
Byron Bay.

Dear Dick,

Thank you very much for your letter, a little bit of appreciation goes a long way. Despite my incredible youth (my Afro-haired assistant from Arnhem Land), I do try to deal with all your problems promptly and compassionately.

Your Mother Theresa of the Computer World.
Denise.

And finally to Brian from Innaloo, W.A. Your letter complaining of the late delivery of your subscription should really have been addressed to the EDITOR rather than yours truly. However, with some trepidation, I consulted the EDITOR in his bunker; and he replied to the effect that one of our couriers unfortunately ran into a Water Buffalo on the way back from the Printers. Many copies of OZAMIGA were severely water dam-aged, resulting in the aforementioned delay. Better luck next edition. Brian.

Yours apologetically.
Denise.

THE OZAMIGA
DICTIONARY OF COMPUTER TERMS

ACCELERATOR

A hardware expansion that speeds up the CPU. An accelerator makes everything move a lot faster, ie. maths-intensive software and some of the higher level graphics programs, like morphing and raytracing programs are more pleasing to use when it takes less time for their allotted tasks.

AGNUS

This is the chip that has the last say when all the other chips want to access the Amiga's ChipRAM. Various versions of Agnus allow the chips to access different amounts of memory. The latest version allows 2 Megabytes of memory to be allocated for use. Each chip has to take it's turn with the ChipRAM.

AMIGADOS

Disk Operating System. AmigaDOS provides the Amiga with basic functions necessary for the computer to work.

ARCHITECTURE

This term refers to what a computer looks like on the inside, rather than it's physical design. The electronic configuration so to speak.

ARCHIVE

An Archive is a compressed data storage system. A number of selected files may be archived(stored) into one file. (eg ** .lha). Archived files are very space efficient but they must be unarchived before use. Some popular Archivers on the Amiga are LHA, LZ, LHArc, Arc, Zoo

AREXX

This is a language a bit like BASIC, but a lot slower. It's advantage is that it can communicate with ARexx compatible programs, swapping data and providing Macro functions for tedious and repetitive operations. ARexx is the interprocess communication language which is supplied with every new Amiga.

ASCII

The standard format for any text storage on the Amiga.

BAUD RATE

This refers to the maximum cps(characters per second) which can be transmitted between two modems. For example; if you connect two modems at 2400 baud, the data will be transmitted at 240 characters per second. Similarly to connect at 9600 baud will give you a maximum of 960 cps. Please take note; these baud rates are only the maximum. If your telephone connection suffers from any interference then your transmission rate may be retarded somewhat. One other point to remember is that if your modem is faster than the one at the other end, you will only be able to receive transmissions at the slower rate.

BLITTER

Block Image Transfer Device used for the transference of large areas of memory from A to B, or the combination of different areas of memory into one single image. Many computer games use the blitter to generate large graphic shapes.

BRIDGEBOARD

An IBM emulating device. A complete PC on a single card that enables the use of standard PC cards and most PC software.

CHIP RAM

Memory accessible by the processor and the custom chips(Agnus, Paula, Denise). Used to store all sounds and graphics. Programs can be stored in chip RAM too, with the sacrifice of speed.

CLI

Command Line Interface. The program which enables you to type commands to AmigaDOS for standard operations like copying and deleting files, or listing a directory. Shell is more commonly used now.

COPPER

Co-processor. This device is built into the Agnus chip and executes instructions at the same time as the processor. Used to generate subtly coloured backgrounds while the screen is being generated

CPU

Central Processing Unit. The chip that makes all the decisions and tells everything else what to do. Custom hardware can interupt the CPU if it's programs are in chip RAM.

DENISE

This is the chip that controls the graphics display modes, eg. Super High-res and productivity etc.

DIP

Dual In-line Package. A memory chip(the original type) which looks like all chips used to before the advent of surface mount technology and space-saving pin outs. A typical DIP is a 256 x 4 DRAM, as used in the A590 hard drive and some accelerator cards. The main problem with these chips is the actual process of inserting them; the pins are easily bent.

ECHOES

Are different 'files' that contain messages from BBS users. Each echo deals with a particular subject so BBS users don't have to trudge through mounds of messages of no intrest to them to find what they do want to know. There are many specific Amiga echoes that travel throughout Australia or around the world. Which means you can communicate with BBS users elsewhere on this globe for the price of a local call. These echoes are public, that is, anyone anywhere in the world can read and write in them. There is another communication system for private mail, called Netmail.

ECS

Enhanced Chip Set. The redesigned custom chip set of the Amiga. First seen as part of the A500+ then A600.

ENV

This is the directory used to store all environmental variables and is usually set up by the Workbench startup-sequence as a directory in the RAM disk.

FAST RAM

Memory that is only accessible by the processor. Programs stored in fast RAM run with more speed than those in chip RAM.

FIDONET

There are Bulletin boards throughout the world that collect messages from their local users, then use the Fidonet system to transmit those messages every night, to their destination at other BBS's. Each Fidonet BBS receives a packet of messages from the board behind it in the chain, stores them for it's own users, adds messages from those users then awaits a call from the BBS ahead of it in the chain to collect the resulting package.

FLOW CONTROL

Sometimes it is useful for the connection rate between your Amiga and modem to be different from that between your modem and the remote modem. We call this flow control. Supposedly to ensure your computer and modem don't lose fragments of data between them. The most common flow control is RTS/CTS, which is hardware controlled, and XON/XOFF controlled by

software. Both the Amiga and your modem must agree on the settings used. These can be set up using both the Amiga preferences and your terminal software settings.

FRAME GRABBER

A device for the capturing of live video data and converting it into computer picture data. The results are usually a lot better than those with a video digitiser, which does the same job, except the video must be paused.

FREEZER CARTRIDGE

A device for the examination of programs while they are running. Usually the cartridge has many built in utilities which help to obscure it's other more sinister uses.

GUI

Graphical User Interface. Basically; when you load workbench on the Amiga you are confronted with the Amiga's GUI. (Windows, Icons, Menus, Pointers) Meaning that control of the computer is largely governed by the mouse and not the keyboard.(ie. Point and click.)

HAM

Hold And Modify. A graphics mode used by the Amiga to display 4096 colours.

HARD DRIVE

Basically a big disk for storing all the programs that you use, including your workbench. A hard drive has a much faster access time than a floppy disk. So a higher rate of productivity is gained.

ICON

Icons are the small pictures which some files and devices are given to help you locate them on the Workbench.

IFF

Interchange File Format. This is a standard way of storing pictures and sounds. Most art programs can load and save pictures in this form. IFF was invented by Electronic Arts and adopted by Commodore.

KICKSTART

This is the name of the set of programs that are run when the Amiga is turned on. The Kickstart also contains many library functions for the use of programmers. Version releases of Kickstart to date are 1.1, 1.2, 1.3, 2.0, 2.04, 2.05, 2.1, 3.0

MENU

A menu is a little list that drops down from the top of the screen usually via the depression of the right mouse button. The menu contains elements known as items and may, in turn, contain further menus.

MORPH

Morphing is a digital effect that is relatively new to the Amiga. It's the process of smoothly altering the shape and colour of an image or scene to create a totally new image. This is a complex procedure in which each part of the original image actually changes colour and position.

NETMAIL

If you want to send private messages to a particular person you should use Netmail. You might find a BBS that doesn't handle Netmail, but most BBS sysops will happily allow the transmission of Netmail, although some ask a fee to do so.

Portfolio

of an *Artist.*
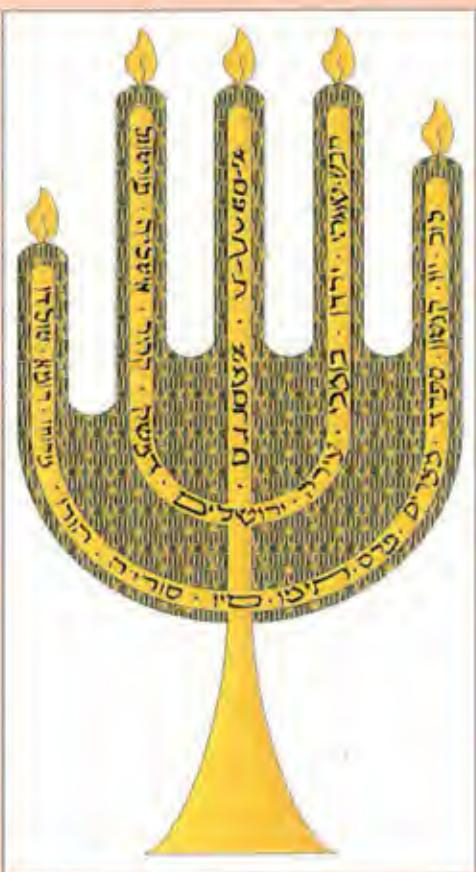
David Jacobs is the editor of the publication "24-bits and pieces". He runs a company called Creations Audio Visual Services and likes working with Deluxe Paint, Opalpaint and Real 3D. David can be contacted on (09) 276-7850